PeerReview Complete v6.5.6500 Owner's Manual

User's and Administrator's Guide

© 2003-2011 SmartBear Software

PeerReview Complete v6.5.6500 Owner's Manual

© 2003-2011 SmartBear Software

All rights reserved. No parts of this work may be reproduced in any form or by any means - graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems - without the written permission of the publisher.

Products that are referred to in this document may be either trademarks and/or registered trademarks of the respective owners. The publisher and the author make no claim to these trademarks.

While every precaution has been taken in the preparation of this document, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

Printed July 2011 in Austin, TX, USA.

Table of Contents

Foreword 0

Part I Big Picture 1

1 PeerReview Complete Overview 1

2 Moving Parts 1

3 Quick Start 3

For System Administrators 3

For Developers 10

4 How Do I? 16

How Do I Get Started? 16

How Do I Do Reviews? 17

How Do I Do Setup or Administrate the Server? 22

5 Contact Us 24

Part II Server Administration 24

1 Installation 24

Database Installation 24

Zero-Configuration 25

MySQL 25

SQL Server 27

Oracle 29

Server Installation 31

Server Upgrades 37

Testing Newer Versions 38

Network Configuration 39

Platform-Specific Notes 40

2 System Administration 41

Licensing 41

Content Storage 44

Backup / Migration 46

Security Considerations 49

LDAP Authentication 50

Configuring HTTPS 55

JMX Monitoring 60

Technical Specifications 61

High Availability Best Practices 63

Diagnostics 65

Logging 67

Troubleshooting 69

3 PeerReview Complete Configuration 72

General Settings 73

User Management 84

Groups 88

Syncing Groups 92

Using Groups for Organizational Hierarchy 94

Using Groups for Projects 95

© 2003-2011 SmartBear Software

Using Groups for Organizational Hierarchy and Projects ... 97
E-Mail Configuration ... 99
Version Control ... 101
Custom Fields ... 104
Role Configuration ... 107
Workflow Configuration ... 112
Triggers ... 115
Notification Templates ... 116
Automatic Links ... 117
Variable Substitution ... 119
Archiving Data ... 122
System Status ... 123

Part III Web User's Guide 124

1 Account Management ... 124
Log in / Log out ... 125
User Preferences ... 127
Notifications ... 134
2 Reviews ... 136
Review Overview ... 136
Action Items ... 139
Creating a Review ... 140
Review Summary Screen ... 148
Reviewing Materials ... 157
Reviewing Text Files ... 157
Reviewing Images ... 162
Reviewing Documents ... 164
Reviewing URLs ... 166
Review Chats ... 167
FAQ's ... 173
3 Searching & Reporting ... 174
Searching ... 174
Reporting ... 176

Part IV Clients 182

1 Installation ... 182
2 Configuration ... 187
Server Connection Configuration ... 187
3 GUI Client ... 189
Main Screen ... 190
SCM Configuration ... 191
Preferences ... 193
Troubleshooting ... 196
4 Command Line Client ... 196
Configuration ... 197
Commands ... 198
Graphical Editor ... 200
Uploading Diffs ... 200
Troubleshooting ... 204
Global Options Reference ... 204
Command-line Reference ... 211

© 2003-2011 SmartBear Software

help ... 213
info ... 213
login ... 214
logout ... 214
set ... 215
addchanges ... 216
addfiles ... 217
addchangelist ... 218
adddiffs ... 219
addsvndiffs ... 220
addardiffs ... 220
addcvsdiffs ... 221
addgitdiffs ... 222
addhgdiffs ... 223
addp4diffs ... 224
addstdiffs ... 225
addvssdiffs ... 226
addversions ... 227
addactivity ... 229
actionitems ... 230
addtrack ... 230
addstream ... 231
browse ... 232
commit ... 232
addp4job ... 233
addurls ... 233
admin ... 234
- syncusers ... 235
- review-xml ... 235
- batch ... 236
- wget ... 237
- config ... 238
  - custom-field ... 238
  - edit ... 238
- review ... 239
  - create ... 240
  - edit ... 241
  - delete ... 242
  - cancel ... 242
  - wait ... 243
  - finish ... 243
  - copy-participants ... 244
  - set-participants ... 244
  - participant ... 245
  - assign ... 245
  - remove ... 246
  - poke ... 246
  - comment ... 247
  - create ... 247
  - redact ... 249
  - defect ... 249
  - create ... 249
  - edit ... 251
  - mark-fixed ... 251

© 2003-2011 SmartBear Software

mark-external ... 252
mark-not-external ... 252
mark-open ... 252
delete ... 253
conversation ... 253
mark-read ... 254
accept ... 255
user ... 256
create ... 256
edit ... 257
author-subscription ... 258
create ... 259
edit ... 259
delete ... 259
file-subscription ... 260
create ... 260
edit ... 261
delete ... 261
trigger ... 262
ensure-review-started ... 262
ensure-reviewed ... 264
ensure-content-reviewed ... 266
ensure-diffs-reviewed ... 268
update-changelist ... 268
create-review ... 270
group ... 272
create ... 272
edit ... 273
delete ... 274
sync ... 274
member ... 276
add ... 276
remove ... 277
changelist ... 277
update-id ... 278

5 Tray Notifier ... 278

6 External Diff Viewer launcher ... 280

7 Print To Review ... 282

8 Visual Studio Add-in ... 286

Part V Version Control Integration 287

1 AccuRev Integration ... 288

GUI Client ... 289

Comparing two Streams ... 292

Command Line Client ... 292

addchanges (for AccuRev) ... 293
addchangelist (for AccuRev) ... 294
addardiffs ... 295
addstream ... 296

2 Borland StarTeam Integration ... 297

GUI Client ... 297

Comparing two dates ... 298

© 2003-2011 SmartBear Software

Comparing two Labels ... 299
Comparing two Revisions ... 300
Command Line Client ... 301
addstdiffs ... 301
3 CVS Integration ... 302
GUI Client ... 303
Comparing two Labels ... 305
Comparing two dates ... 306
Command Line Client ... 307
addchanges (for CVS) ... 308
addcvsdiffs ... 309
commit (for CVS) ... 310
4 Git Integration ... 311
GUI Client ... 312
Comparing two branches ... 315
Command Line Client ... 315
addchanges (for Git) ... 315
addgitdiffs ... 317
5 IBM CMVC Integration ... 318
GUI Client ... 318
Command Line Client ... 318
addchangelist (for CMVC) ... 319
addversions (for CMVC) ... 319
addtrack ... 321
6 IBM Rational ClearCase Integration ... 322
GUI Client ... 324
Command Line Client ... 328
addchanges (for ClearCase) ... 329
addchangelist (for ClearCase) ... 331
addversions (for ClearCase) ... 332
addactivity ... 334
commit (for ClearCase) ... 335
7 IBM Rational Synergy Integration ... 336
GUI Client ... 336
Command Line Client ... 339
addchangelist (for Rational Synergy) ... 340
addversions (for Rational Synergy) ... 341
8 Mercurial Integration ... 343
GUI Client ... 344
Command Line Client ... 346
addchanges (for Mercurial) ... 346
addhgdiffs ... 348
9 Microsoft Team Foundation Server Integration ... 349
GUI Client ... 349
Command Line Client ... 352
addchanges (for Team Foundation Server) ... 353
addchangelist (for Team Foundation Server) ... 354
commit (for Team Foundation Server) ... 355
10 Microsoft Visual SourceSafe Integration ... 356
Command Line Client ... 356
addvssdiffs ... 357

© 2003-2011 SmartBear Software

11 MKS Source Integrity Integration 358
GUI Client 358
Command Line Client 361
addchanges (for MKS Source) 362
addchangelist (for MKS Source) 363
addversions (for MKS Source) 364
12 Perforce Integration 366
Perforce Server Integration 367
GUI Client 369
Comparing files in a Branch 372
Comparing two Labels 372
Comparing two dates 373
Command Line Client 374
addchangelist (for Perforce) 375
addp4diffs 376
addversions (for Perforce) 377
commit (for Perforce) 379
addp4job 379
syncusers 380
update-id 380
P4V / P4Win Integration 381
Perforce Server Triggers 383
ensure-review-started (for Perforce) 383
ensure-reviewed (for Perforce) 385
ensure-content-reviewed 386
update-changelist 387
13 SourceGear Vault Integration 390
GUI Client 392
Command Line Client 394
addchanges (for Vault) 395
14 Subversion Integration 396
Subversion Server Integration 397
GUI Client 399
Comparing two Revisions 402
Comparing two branches / tags 403
Comparing two dates 404
Command Line Client 405
addchanges (for Subversion) 406
addchangelist (for Subversion) 408
addsvndiffs 408
commit (for Subversion) 409
Subversion Server Hooks 410
ensure-review-started (for Subversion) 411
ensure-reviewed (for Subversion) 412
create-review 413
ensure-review-started (for Subversion) 415
ensure-reviewed (for Subversion) 416
create-review 417
15 Surround SCM Integration 419
GUI Client 419
Command Line Client 421
addchanges (for Surround SCM) 422

© 2003-2011 SmartBear Software

addchangelist (for Surround SCM) ... 423
addversions (for Surround SCM) ... 424

Part VI External Integrations 426

1 Custom Reports ... 426
Database Schema ... 427
Example SQL: Participant Custom Fields ... 435
Example XPath and XSL ... 441
2 Bug-Tracking Integration ... 442
3 Scripting ... 443
Mirroring Defects to an external issue-tracker ... 445
Syncing users from Perforce ... 448
Conversation with Eliza ... 449

Part VII Techniques & Best Practices 454

1 Metrics: Definitions ... 455
2 Metrics: Analysis ... 456
3 Multiple Change Changelists ... 458
4 Tips and Tricks ... 458

Part VIII Appendices 460

1 Appendix A: Known Issues ... 460
2 Appendix B: Version History ... 462
Version 6 ... 463
Version 5 ... 472
Version 4 ... 485
Version 2 ... 502
Version 1 ... 512
Version 0/Alpha ... 519
3 Appendix C: Java VM Options ... 525

Index 528

© 2003-2011 SmartBear Software

1 Big Picture

This section will get you acquainted with PeerReview Complete and go through the product overview [1], components [1], and a quick start guide [3].

1.1 PeerReview Complete Overview

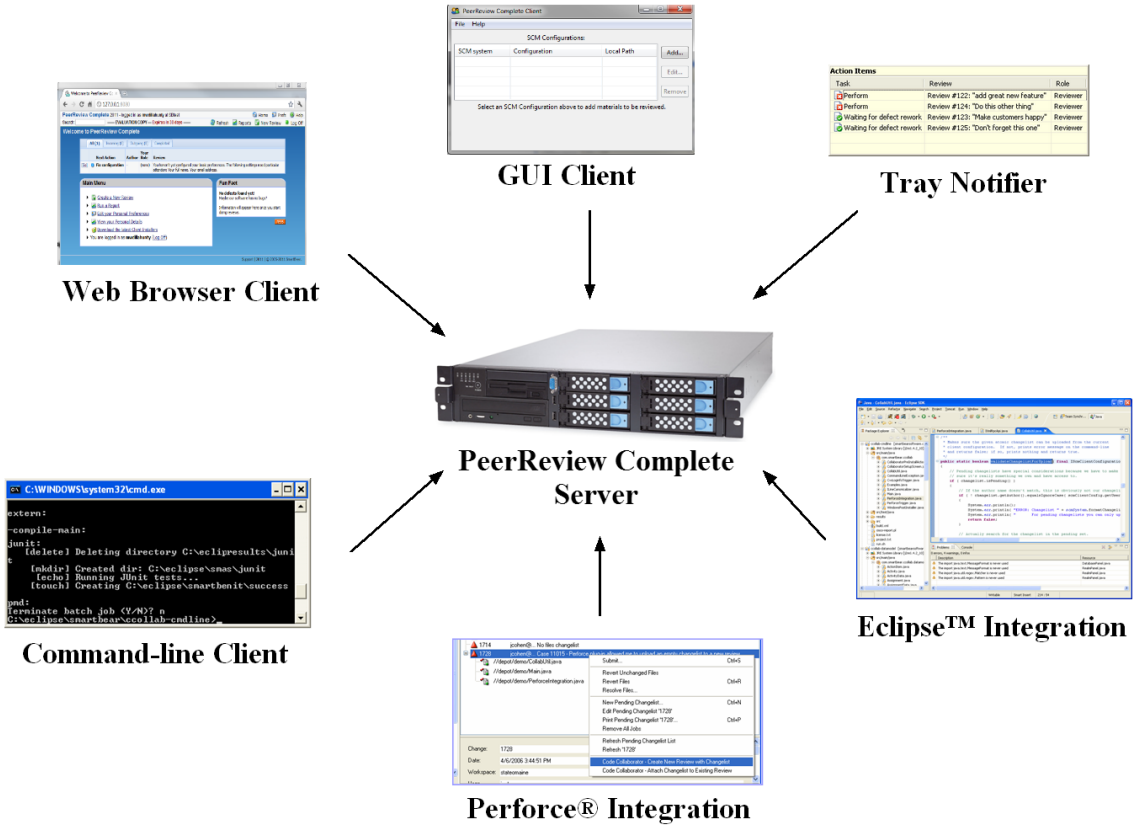
PeerReview Complete is a peer review system supporting:

- Multiple, simultaneous reviewers
- Workflow supporting reviewers/authors separated by many time zones
- Version control integration
- Defect-tracking with severity, type, classification, checklists, and external issue-tracker integration
- Full-featured metrics, reports, and data-export
- Project- and role-based rules and reporting
- Web-based, rich-client, and command-line cross-platform clients
- Command-line API for integrations, extensions, automations, and triggers

1.2 Moving Parts

PeerReview Complete includes a server component, and a variety of clients, 3rd-party integrations, and client protocols.

© 2003-2011 SmartBear Software



PeerReview Complete Server 24

As with most enterprise-class software systems, a server process acts as the hub, manager, and controller of information. The server has a web-based user interface where users and administrators can do everything — create and perform reviews, configure personal and system-wide settings and run reports. The server uses an external database 24 to store all data and configuration.

Web Browser Client 124

The web browser client is used by entering an external URL in a web browser to connect to the PeerReview Complete server. This is where most of the review will take place, where you edit, view, configure the reviews. The web interface supports Internet Explorer, Safari, Firefox and Chrome.

Command-line Client 196

Developers will typically install the cross-platform client. This tool includes a Command Line Client that lets you upload local files (and file-changes) into new and existing reviews. The Command Line Client also includes scripting commands for implementing custom behavior and integrating with external systems.

There are many reasons why you might want to integrate PeerReview Complete with other systems. An issue-tracker integration point might let you synchronize PeerReview Complete "defects" with issue-tracker "issues," or you might want to mirror review data (metrics/comments/file-differences) into the associated ticket. A reporting integration point might let you mirror PeerReview Complete metrics into your existing reporting system (examples: defects/kLOC, defects/man-hour, kLOC/man-hour, number of defects found of different types or severities, etc).

© 2003-2011 SmartBear Software

GUI Client 189
The cross-platform client install includes a graphical client to complement the web-based user interface already provided by the server. The GUI client is a cross-platform client available for Windows, Mac, and Unix/Linux users.

Tray Notifier 278
In addition to uploading files, the GUI Client gives you a taskbar icon that updates to show you whether you have any pending tasks in PeerReview Complete. This is called the Tray Notifier and is available to Windows and Unix/Linux users. This allows users to easily access pertinent reviews and alerts the user of any new activity in reviews without being actively involved in the web browser client.

Perforce® Integration 366
Perforce users will probably want to install the Perforce Client Integration tools. These are included in the client installer 182.

Integration with P4V and P4Win lets users upload changelists into new or existing reviews just by right-clicking on the changelist. This works on both "pending" and "submitted" changelists.

We also supply a special tool for use as a Perforce server trigger. For example, you can use this to enforce a rule like "Every submit on this branch requires a review." You can also use this to automatically upload all submitted changelists into PeerReview Complete so that you can review files after they have been checked in. This can be especially useful with off-shore development groups.

1.3 Quick Start

This section will briefly guide you through the steps of getting started with PeerReview Complete. It is only intended for users who want to jump right into PeerReview Complete and start using the basic features. If you'd like more detailed instructions on how to use PeerReview Complete, please visit the appropriate sections throughout the Owner's Manual.

We've divided this section into two guides. The Quick Start for System Administrators 3 will show you how to install and configure the server component. The Quick Start for Developers 10 will describe the steps needed to install the client component and start your first review.

1.3.1 For System Administrators

1. **Download the PeerReview Complete Server**
   In order for your users to use PeerReview Complete, you must first install and configure the server component. Download the “Server Installer” listed under your platform from the Download page: http://downloads.smartbear.com.

2. **Installation Wizard**
   Once you’ve downloaded the server installer, open it to begin the installation wizard. The wizard guides you through the steps below:

   a. **Welcome Page**
      This screen welcomes you to the installation process. Click *Next* to proceed.

   b. **License Agreement**

© 2003-2011 SmartBear Software

Read the license agreement, and select *I accept the agreement* before clicking *Next*.

c. Destination Directory

Choose the folder where you would like the PeerReview Complete server to be installed. Make sure that the directory is writable by the server and that there are at least 5 gigabytes of space available in the directory.

d. Start Menu Folder

Select the Start Menu folder in which you'd like to create the PeerReview Complete shortcut.

e. Configure Webserver

Enter an open port number that does not conflict with any existing services. Port 8080 is the standard port we default to. When you click *Next*, the installer attempts to connect to this port. If the port is in use, a warning message appears and you must specify a different port or free up the desired port.

f. Database Type

Choose one of the following database types:

- *Embedded* – If you are trying out PeerReview Complete, we recommend that you select Embedded to keep things simple. This choice does not require a prior setup. However, if you're ready to use PeerReview Complete in production, we require that you use a full-featured database instead of the Embedded database. Using the Embedded database with more than a few reviews causes complications and may lead to irreversible loss of data. But don't worry; we have a built-in database migration tool, so you can easily migrate from the Embedded database to a full-featured database later.
- *MySQL, SQL Server, or Oracle* – These are the three full-featured databases we support in PeerReview Complete. If you do not use any of these, we recommend MySQL as it's a free, open-source database. (It's also important to note that if you are planning to use CodeReviewer, only MySQL is supported.) If you choose one of these databases, you are taken to a screen where you can configure a connection for the PeerReview Complete server to use for accessing the database server. You must have your database already installed and set up in order to correctly configure this. For more details, visit the Database page 24. If you do not have the database set up, you can always choose Embedded for now and migrate later. If you have your database ready, fill out the configuration details using the database username and password for the Username and Password.

g. Authentication Type

Choose one of the following authentication types:

- *Internal* – PeerReview Complete maintains usernames and passwords in its own database.
- *LDAP* – You can use LDAP authentication, which won't require you to maintain the PeerReview Complete usernames and passwords. If you choose this option, the next screen lets you configure your LDAP integration. Since this configuration is a bit trickier, please visit the LDAP section 50 of our documentation for full details.

h. Completed Page

This page appears once the server is installed. Click *Finish*.

3. Log into the Web Browser Client

After you finish with the installation, a web browser opens automatically, pointed at your installed

© 2003-2011 SmartBear Software

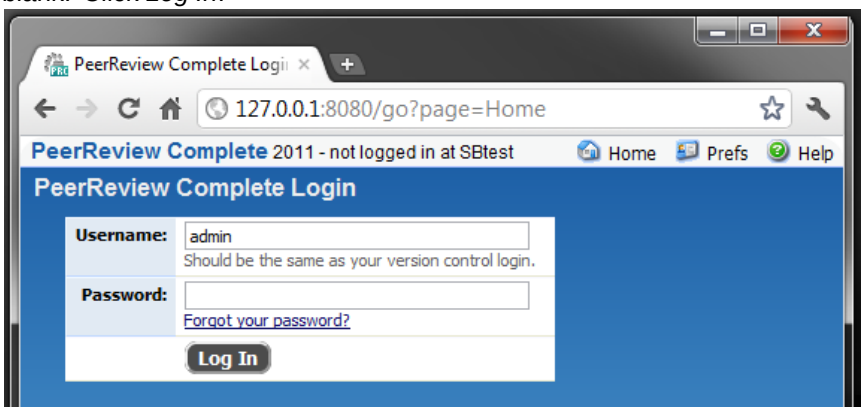
server. This page is part of the PeerReview Complete web browser client, which you will use to administer your PeerReview Complete server.

a. Database Initialization Required

When opening the web browser for the first time after configuring the PeerReview Complete server, "Database Initialization Required" is displayed. Click *Initialize Database* to continue.

b. Log In

- *Internal Authentication* – If you have chosen Internal Authentication earlier in the installation process, specify "admin" as the Username, and leave the Password field blank. Click *Log In*.

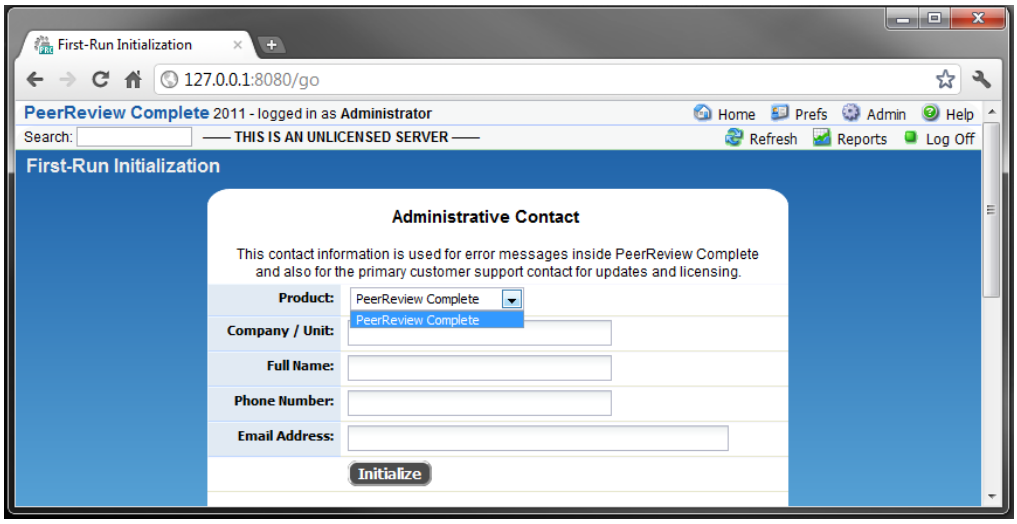


- *LDAP Authentication* – For LDAP authenticated installations, the admin username and password will be the username and password you have provided earlier in PeerReview Complete installation process when configuring your LDAP integration. Please view our LDAP page 50 for more information.

4. First-Run Initialization

Once you have logged in as the admin, the next page asks you to provide necessary information to set up the PeerReview Complete server.

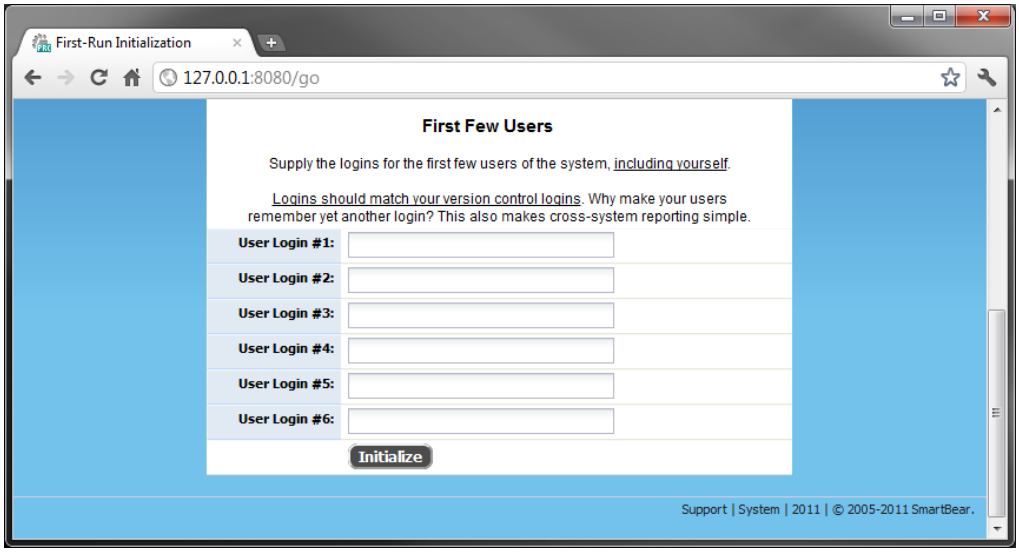
- *Administrative Contact* – This section allows you to choose the product type and provide the administrator's contact information.



© 2003-2011 SmartBear Software

In the first field, Product, use the drop down menu to choose which product you'd like to use, PeerReview Complete or CodeReviewer. Note: If you're using PeerReview Complete, you will have no other Product options. All products have the same goal, to make your peer review process easier and more effective, but PeerReview Complete offers more functionality regarding customization, configuration, etc. To view all differences between PeerReview Complete and CodeReviewer, visit our website, http://smartbear.com/codecollabcodereviewer.php. In the remaining fields under this section, fill in the company name, the PeerReview Complete administrator's full name, phone number, and e-mail address. SmartBear only uses this information to contact you when there is a critical issue with your PeerReview Complete installation; we will not sell or share this information. The contact information is primarily there for your users to know who to contact if they are having issues with PeerReview Complete.

- *First Few Users* – The next section helps you get started on populating your user list.

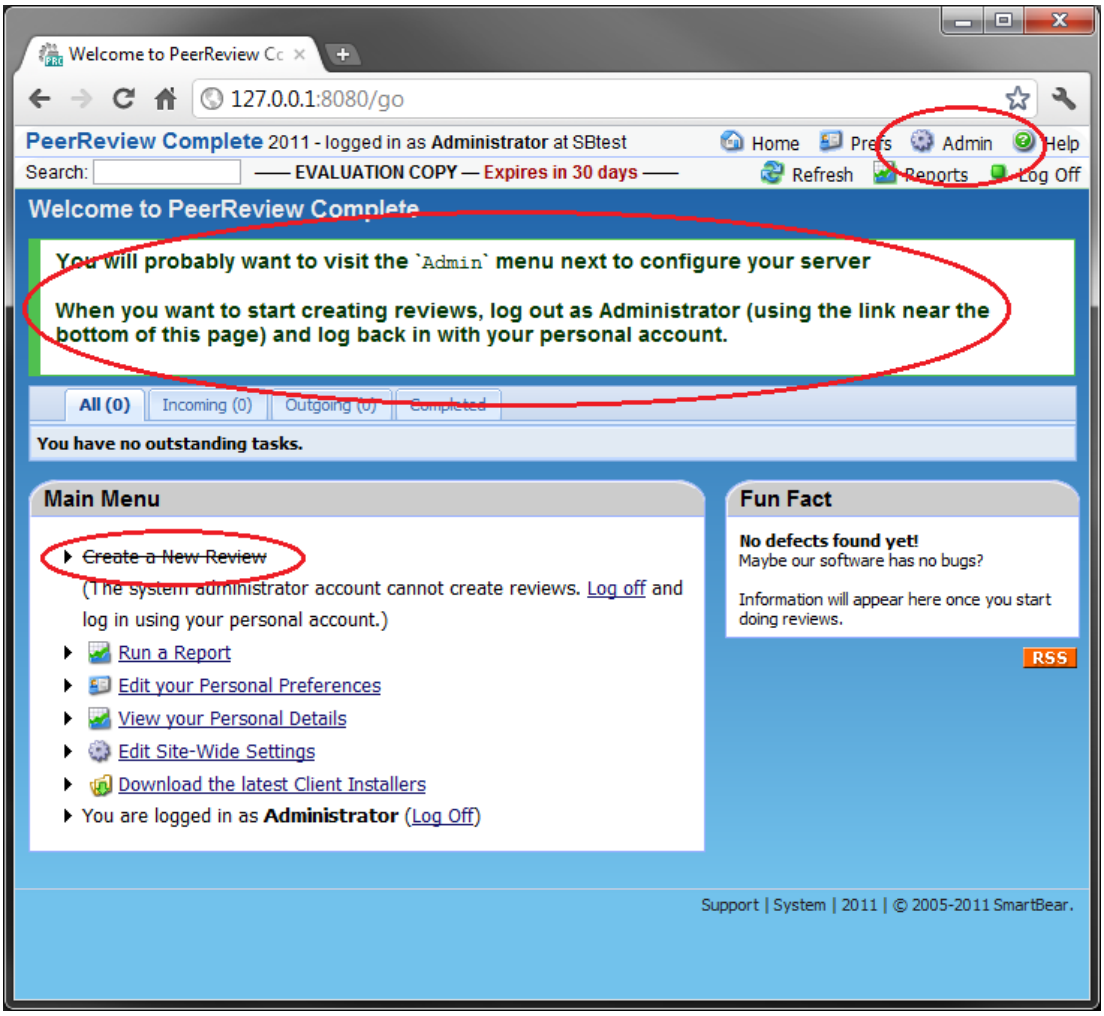

First-Run Initialization
127.0.0.1:8080/go
First Few Users
Supply the logins for the first few users of the system, including yourself.
Logins should match your version control logins. Why make your users remember yet another login? This also makes cross-system reporting simple.
User Login #1:
User Login #2:
User Login #3:
User Login #4:
User Login #5:
User Login #6:
Initialize
Support | System | 2011 | © 2005-2011 SmartBear.

You should supply at least two usernames for this section. Typically, the admin login is used only for administrative purposes, so we recommend that you enter a personal username for yourself to use for reviewing purposes. We also recommend that the PeerReview Complete usernames match your version control usernames for simplicity. Click *Initialize* once you have entered the usernames. This section only appears if you have selected Internal authentication. LDAP authenticated installations do not have this option because your usernames are already set.

5. Home Page

After the first-run initialization, you're taken to the PeerReview Complete Home page. This page looks exactly like your average PeerReview Complete user's home page with a few exceptions: a green notification box reminding you to configure your server, the Admin button, and a disabled Create a New Review link in the main menu (since by default, PeerReview Complete does not allow administrators to create reviews).

© 2003-2011 SmartBear Software

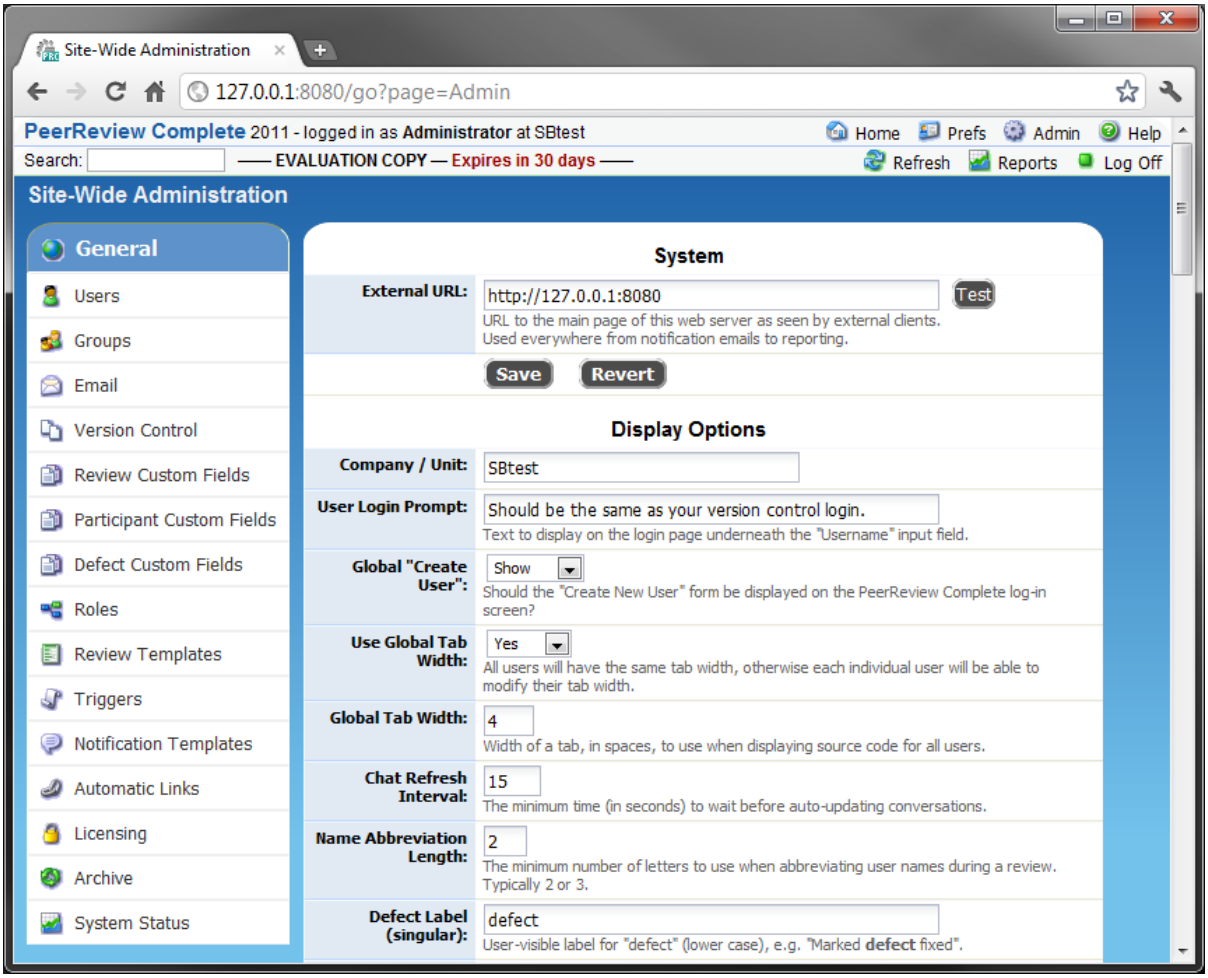

Welcome to PeerReview Cc
127.0.0.1:8080/go
PeerReview Complete 2011 - logged in as Administrator at SBtest
Home
Prefs
Admin
Help
Search:
EVALUATION COPY — Expires in 30 days
Refresh
Reports
Log Off
Welcome to PeerReview Complete
You will probably want to visit the `Admin` menu next to configure your server
When you want to start creating reviews, log out as Administrator (using the link near the bottom of this page) and log back in with your personal account.
All (0)
Incoming (0)
Outgoing (0)
Completed
You have no outstanding tasks.
Main Menu
Create a New Review
(The system administrator account cannot create reviews. Log off and log in using your personal account.)
Run a Report
Edit your Personal Preferences
View your Personal Details
Edit Site-Wide Settings
Download the latest Client Installers
You are logged in as Administrator (Log Off)
Fun Fact
No defects found yet!
Maybe our software has no bugs?
Information will appear here once you start doing reviews.
RSS
Support | System | 2011 | © 2005-2011 SmartBear.

As thePeerReview Complete administrator, you'll want to familiarize yourself with the *Admin* button at the top right, so click the *Admin* link.

6. Admin Section

Once you click the *Admin* link, the browser directs you to the Admin section. The Admin section is where you can configure the settings to your liking.

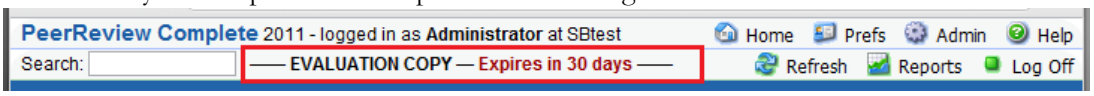
© 2003-2011 SmartBear Software


Site-Wide Administration
127.0.0.1:8080/go?page=Admin
PeerReview Complete 2011 - logged in as Administrator at SBtest
Home Prefs Admin Help
Search:
— EVALUATION COPY — Expires in 30 days —
Refresh Reports Log Off
Site-Wide Administration
General
Users
Groups
Email
Version Control
Review Custom Fields
Participant Custom Fields
Defect Custom Fields
Roles
Review Templates
Triggers
Notification Templates
Automatic Links
Licensing
Archive
System Status
System
External URL:
http://127.0.0.1:8080
Test
URL to the main page of this web server as seen by external clients.
Used everywhere from notification emails to reporting.
Save
Revert
Display Options
Company / Unit:
SBtest
User Login Prompt:
Should be the same as your version control login.
Text to display on the login page underneath the "Username" input field.
Global "Create User":
Show
Should the "Create New User" form be displayed on the PeerReview Complete log-in screen?
Use Global Tab Width:
Yes
All users will have the same tab width, otherwise each individual user will be able to modify their tab width.
Global Tab Width:
4
Width of a tab, in spaces, to use when displaying source code for all users.
Chat Refresh Interval:
15
The minimum time (in seconds) to wait before auto-updating conversations.
Name Abbreviation Length:
2
The minimum number of letters to use when abbreviating user names during a review.
Typically 2 or 3.
Defect Label (singular):
defect
User-visible label for "defect" (lower case), e.g. "Marked defect fixed".

The box to the left of the screen is the Admin menu, which contains a number of helpful links that guide you through different configuration sub-categories. See the Other Configurations 10 section of this document for more information regarding specific configurations.

7. Licensing

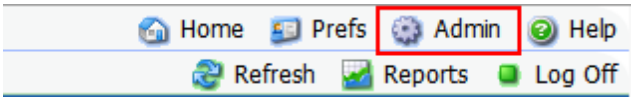
If you have an external internet connection, PeerReview Complete will attempt to contact our licensing server so that we can set you up with a 30-day trial license. Your licensing status is denoted by a description in the top toolbar to the right of the search bar.


PeerReview Complete 2011 - logged in as Administrator at SBtest
Home Prefs Admin Help
Search:
— EVALUATION COPY — Expires in 30 days —
Refresh Reports Log Off

If you do not have an external connection, or if a firewall prevents access to our licensing server, you need to manually obtain a license code. The following instructions also apply if you want to obtain and enter a permanent license code.

a. Admin

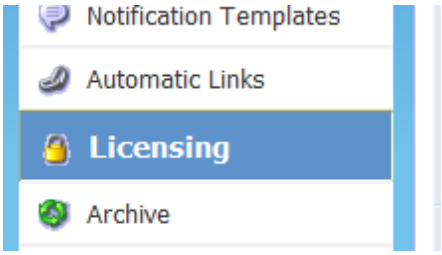
Go to the Admin section. If you're following along with this document, you should already be in the Admin section. If not, click the *Admin* link at the top right.



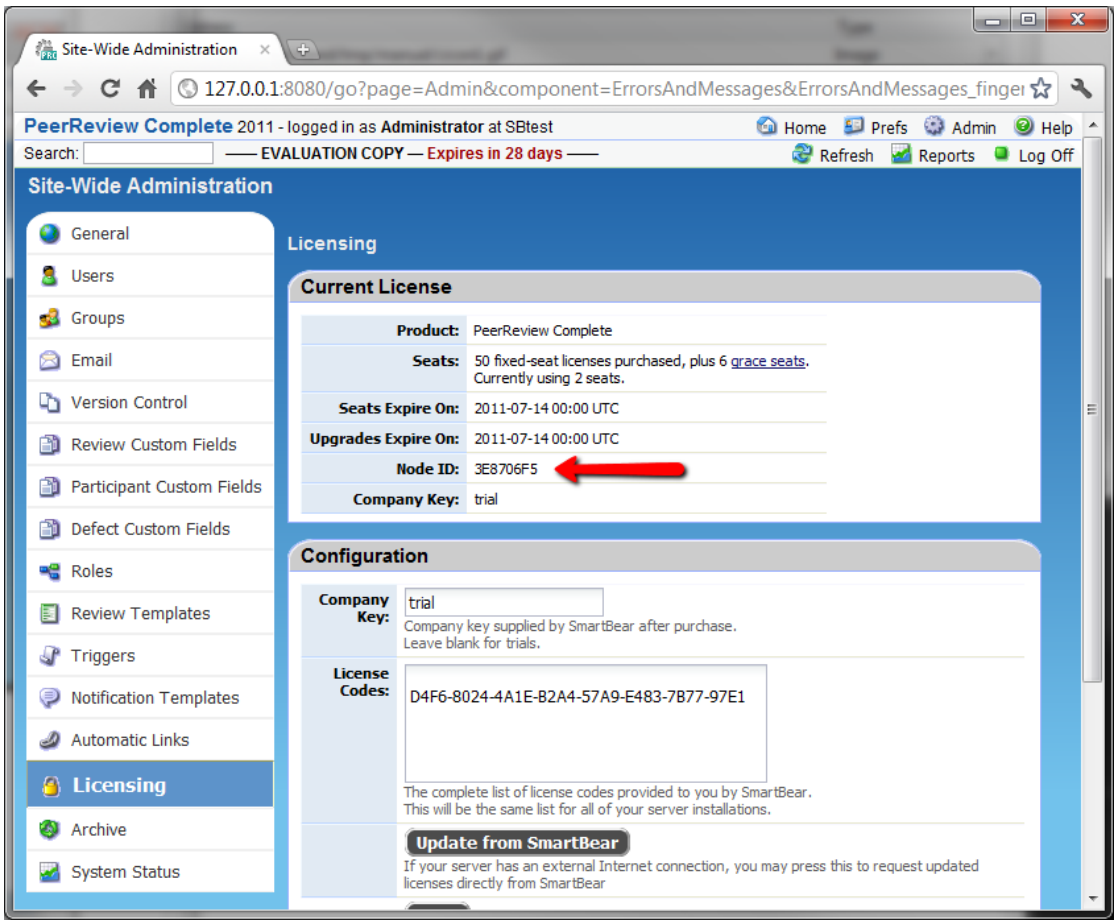
© 2003-2011 SmartBear Software

b. Licensing

Click the *Licensing* link near the bottom of the Admin menu that is located to the left.



This page gives you information regarding the licensing, such as when the license expires, when the support and upgrades expire, how many seats you have, etc.


Site-Wide Administration
127.0.0.1:8080/go?page=Admin&component=ErrorsAndMessages&ErrorsAndMessages_finger
PeerReview Complete 2011 - logged in as Administrator at SBtest
Home Prefs Admin Help
Search:
—— EVALUATION COPY — Expires in 28 days ——
Refresh Reports Log Off
Site-Wide Administration
General
Users
Groups
Email
Version Control
Review Custom Fields
Participant Custom Fields
Defect Custom Fields
Roles
Review Templates
Triggers
Notification Templates
Automatic Links
Licensing
Archive
System Status
Licensing
Current License
Product: PeerReview Complete
Seats: 50 fixed-seat licenses purchased, plus 6 grace seats. Currently using 2 seats.
Seats Expire On: 2011-07-14 00:00 UTC
Upgrades Expire On: 2011-07-14 00:00 UTC
Node ID: 3E8706F5
Company Key: trial
Configuration
Company Key: trial
Company key supplied by SmartBear after purchase. Leave blank for trials.
License Codes: D4F6-8024-4A1E-B2A4-57A9-E483-7B77-97E1
The complete list of license codes provided to you by SmartBear. This will be the same list for all of your server installations.
Update from SmartBear
If your server has an external Internet connection, you may press this to request updated licenses directly from SmartBear

c. Node ID

Find the 8-character Node ID under "Current License". The Node ID is unique to every installation, and the Node ID is tied to every license code we generate, so it is important to send your unique Node ID to your Account Manager to activate your license. If you do not know who your Account Manager is, just send the Node ID to sales@smartbear.com.

d. Company Key and License Code

© 2003-2011 SmartBear Software

After you send us the Node ID, you will receive an e-mail with a company key and license code. The company key is "trial" for temporary licenses and a word or phrase for permanent licenses. The license code will be a 32-character code. Enter the company key and the license code in their respective fields, and click *Save*.

8. Other Configurations

Most of the settings are self-explanatory, and if you need more detailed explanations, visit the Configuration section [73]. But to get started, we'll briefly introduce you to a few important sub-categories and settings. Remember, you can navigate through different sub-categories by using the yellow Admin menu in the Admin section.

a. General [73]

- External URL – Enter the full URL to the main page of the web server on which PeerReview Complete was installed, as it would be seen by external clients.
- Global "Create User" – When you're starting out, and you don't want to manually create usernames for all users, mark this field as "Show" so the users can create their own username and password. Mark it as "Hide" if you want to control the creation of users and usernames.
- Access Restrictions – Control who has access to reviews, reports, subscriptions, and systems information.
- Review Process Options – Control what users can or cannot do during the review process.
- Bug-Tracking Integration – Add the URL for creating a new bug in your bug tracking system here. See the Bug-Tracking page [442] for more information.

b. Users [84] – Create users, view user statistics, and edit user accounts.

c. E-mail [99] – You must configure this section in order for PeerReview Complete to send out e-mail notifications of review activity.

d. Review and Defect Custom Fields [104] – Create your own fields for users to annotate defects and reviews with necessary information.

e. Role Configuration [107] – Specify the rights and privileges of review participants.

f. Triggers [115] – Execute an external script or application when certain events occur.

g. System Status [123] – The "Peak Usage" chart plots the maximum number of simultaneous users, measured by day: This feature is most useful when determining the number of floating seats necessary to handle your PeerReview Complete usage.

1.3.2 For Developers

1. Download the Client Software

You'll most likely need a PeerReview Complete client if you plan on uploading files from your Version Control System (VCS). Download the "Client Installer" listed under your platform from the Download page: http://downloads.smartbear.com.

The Client Installer includes:

© 2003-2011 SmartBear Software

- GUI Client 189
- Tray Notifier 278
- Command Line Client 196
- External Diff Viewer launcher 280
- Perforce P4V/P4Win plug-ins 381
- VCS triggers

If you use Visual Studio, we have a Visual Studio plug-in available; visit the Visual Studio page 286 for installation instructions.

2. Installation Wizard for the Client Installer

Once you've downloaded the client installer, open it to begin the installation wizard. The wizard guides you through these steps:

a. Welcome Screen

This screen welcomes you to the installation process. Click *Next* to proceed.

b. License Agreement

Read the license agreement, and select *I accept the agreement* before clicking *Next*.

c. Select Destination Directory

Choose the folder where you would like the PeerReview Complete client installed. You need at least 51.1 MB of disk space.

d. Select Start Menu Folder

Select the Start Menu folder in which you'd like to create the PeerReview Complete shortcut. This option is only available for Windows.

e. Configure Connection to the PeerReview Complete Server

Provide the PeerReview Complete Server URL, username, and password. Remember to specify the protocol (e.g. http:// or https://) and port number, and path. If you do not know this, ask your PeerReview Complete system administrator.

Note: This step assumes your PeerReview Complete server component has already been installed (see Getting Started with the PeerReview Complete Server 16 for instructions). If it has not, you can skip this step by clicking Next*, then* Continue *in the pop-up window that appears. You can fill this information in later in the Preferences menu of the PeerReview Complete client, but remember to do that before trying to use the client with the server.*

f. Select Additional Tasks

Choose whether to run the Tray Notifier on Startup. This option is only available for Windows.

Note: You may find this option very helpful. Through an easily visible icon on the lower right hand of your screen, the Tray Notifier provides easy access to current reviews and

© 2003-2011 SmartBear Software

to notifications that alert you to take necessary review actions.

You may also see a second additional task here if the installer detects P4V or P4Win (the Perforce GUI clients) installed. Choose whether to "Configure Addons to Perforce Visual Tools". If selected, the installer automatically installs the PeerReview Complete integration into those tools.

g. Finish

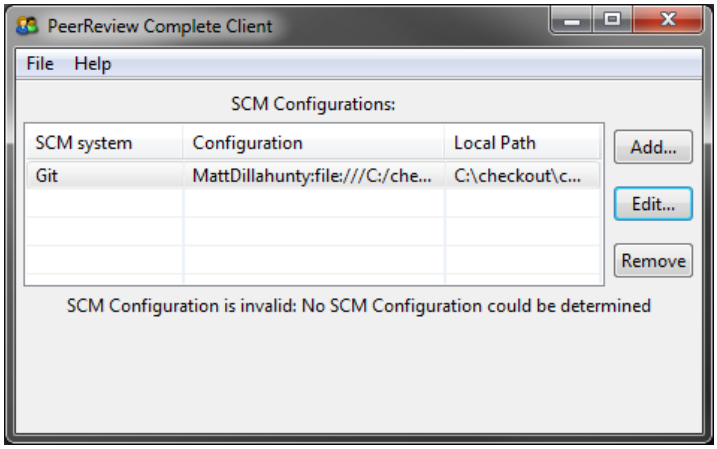
Click *Finish*. You're done installing the client!

3. Configure the Client for your VCS

Once you connect PeerReview Complete to your Version Control System software, the client can gather changed files for you to easily send for review.

Note: You can find a list of supported VCSs here 287.

- GUI – Open up the PeerReview Complete GUI Client. On Windows, you'll find it in the Start -> Programs menu. On MAC OS X, it should be under `Applications -> peerreview_client -> collabgui`. On Linux, `/opt/peerreview_client`. Click *Add…* on the right side of the client.



Use the Configuration wizard to enter the location of your local source into the *Local Path* field at the top, or choose the VCS of your choice.

Note: Visual Source Safe (VSS) is not supported by the PeerReview Complete GUI Client – for more information please refer to the VSS page 356.

- Command-Line – For instructions, visit the Command Line Client page 196.

4. Log into PeerReview Complete Web Browser Client

Open a web browser, and go to the PeerReview Complete server URL provided by your system administrator. Enter your login name and password. (The username and password is typically supplied by the administrator, but if your administrator allows user created logins, a "Create New Account" form appears on the login page where you can create your own login name and

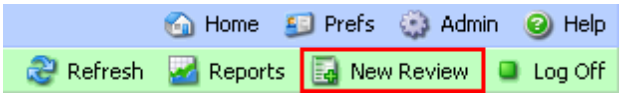
© 2003-2011 SmartBear Software

password.) Once you log in, you can check your Action Items to see if you have any assigned reviews. You can also edit *Preferences* by clicking *Prefs* in the menu bar on the top right of the screen. For more information on User Preferences, go to our User Preferences page 127.

5. Create a Review as an Author

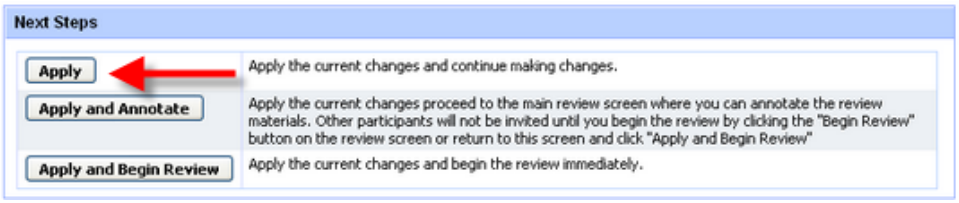
You have several options for creating a review: the Web Browser client, the GUI Client, the Command Line Client, and Perforce users have the additional option of using P4V or P4Win. To ease you into the process, we'll start with the Web Browser client.

a. To make a new review, click *New Review* in the menu bar at the top right.

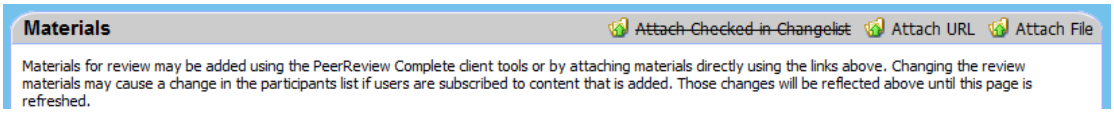


b. Fill in the basic information and participants in the Create a Review screen.

c. Click *Apply*. This step is necessary; PeerReview Complete does not allow you to attach any files to a review unless you click *Apply* first.


Next Steps
Apply
Apply the current changes and continue making changes.
Apply and Annotate
Apply the current changes proceed to the main review screen where you can annotate the review materials. Other participants will not be invited until you begin the review by clicking the "Begin Review" button on the review screen or return to this screen and click "Apply and Begin Review"
Apply and Begin Review
Apply the current changes and begin the review immediately.

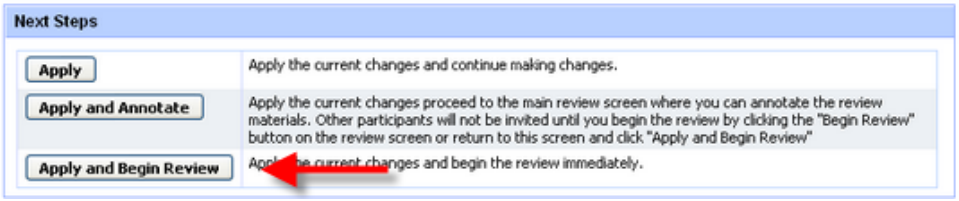
d. The Web Browser client allows you to attach supporting documents and changelists under the Materials section of the Create a Review screen.


Materials
~~Attach Checked-in Changelist~~
Attach URL
Attach File
Materials for review may be added using the PeerReview Complete client tools or by attaching materials directly using the links above. Changing the review materials may cause a change in the participants list if users are subscribed to content that is added. Those changes will be reflected above until this page is refreshed.

- PDF – Click *Attach File* and enter the file path or use the *Browse* button to find files on your local hard drive. You may use the Description field to add any notes regarding your file.
- URL – Click *Attach URL* and enter the link you'd like to review.
- Changelists – If your PeerReview Complete system administrator has configured the PeerReview Complete server with any Version Control System (VCS) servers, you may attach a changelist associated with that VCS by clicking *Attach Checked-in Changelist*. Just choose the VCS and enter the unique changelist number. If your PeerReview Complete server has not been configured with a VCS, the link will be unavailable, denoted by ~~*Attach Checked-in Changelist*~~. In this case, you can use theGUI Client to add files to the review – see step 6 below.

e. Click *Apply and Begin Review* to start the review and send out notifications to reviewers.

© 2003-2011 SmartBear Software

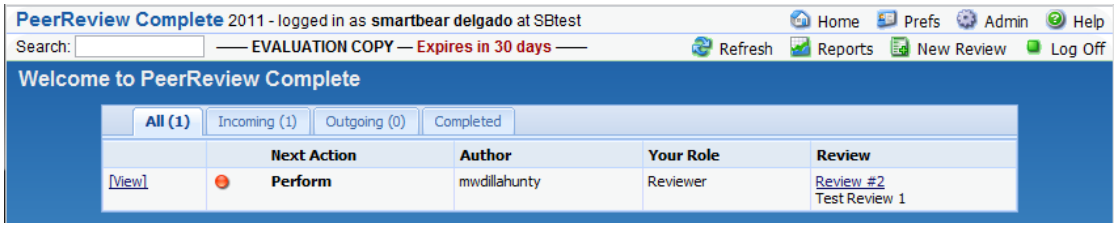

Next Steps
Apply
Apply the current changes and continue making changes.
Apply and Annotate
Apply the current changes proceed to the main review screen where you can annotate the review materials. Other participants will not be invited until you begin the review by clicking the "Begin Review" button on the review screen or return to this screen and click "Apply and Begin Review"
Apply and Begin Review
Apply the current changes and begin the review immediately.

f. Most PeerReview Complete users do not use the Web Browser client for attaching source files, reserving it for adding supporting documents. Instead, most of our users opt to use the GUI Client and/or the Command Line Client for adding source files to a review. The extensive integration these clients provide with Version Control Systems makes them the natural choice for selecting source files to add to a review. To add review materials via the GUI Client or Command Line Client:

- GUI – Open the GUI Client and click on the VCS of your choice. Use the *Add to Review* buttons at the bottom to add materials to an existing or new review. As these buttons will differ according to your Version Control System, we encourage you to visit our GUI Client section 190.
- Command-Line – For instructions, visit the Command Line Client section 196.

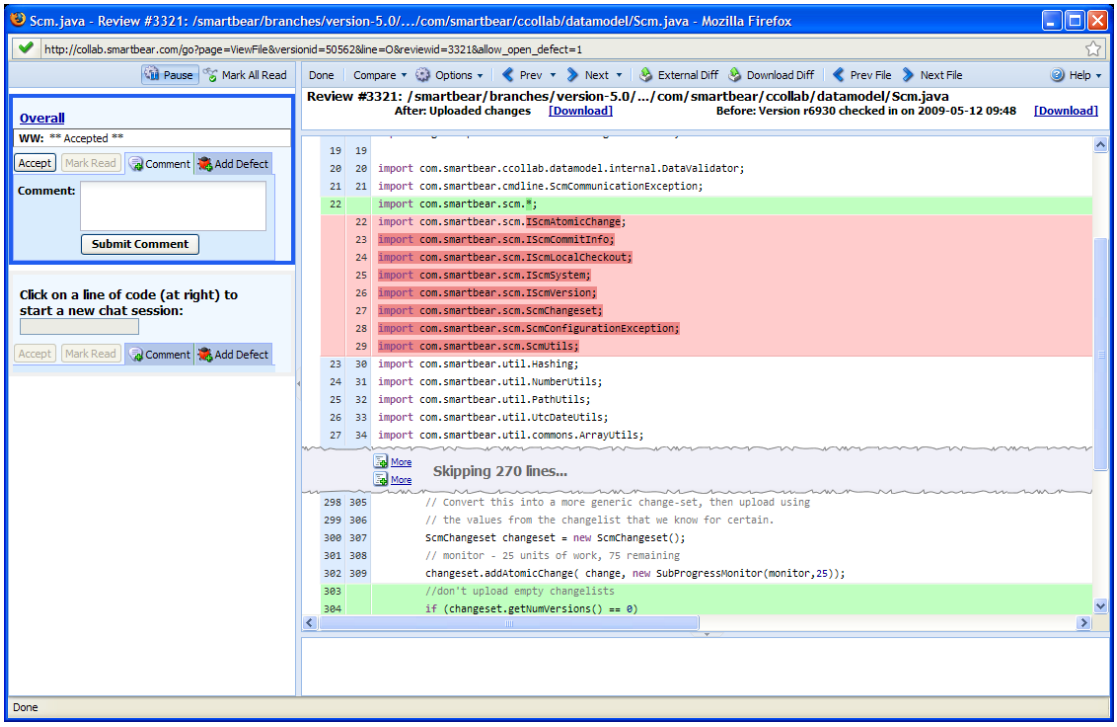
6. Add Comments and Defects as a Reviewer

When someone invites you to a review, you receive an email notification and you also see the review in your Action items list. Click it to open it.



Under the Materials section, click a file. The External Diff Viewer launcher will open, and you will be able to add comments and defects here.

© 2003-2011 SmartBear Software



- Comment – Click on the line (in text files) or area (in supporting docs) you want to comment on, and type your comment. Comments are threaded into a conversation that stays tied to a specific line, and unread comments are marked in yellow.

  Tip: You can click Accept *or* Mark Read *to clear the unread comments.* Mark Read *simply indicates that you have already seen the comment, which may be helpful if you don't want to re-read the same comments when you view a file multiple times.* Accept *does not have an explicit meaning in PeerReview Complete; it is used differently in different environments. You can use this feature however you think best suits your peer review environment. Once this button is used, a green checkmark will appear to notify participants that it has been* Accepted*." (For more information on the* Accept *button, read our blog entry here: http://blog.smartbear.com/the_smartbear_blog/2008/11/what-does-the-a.html)*

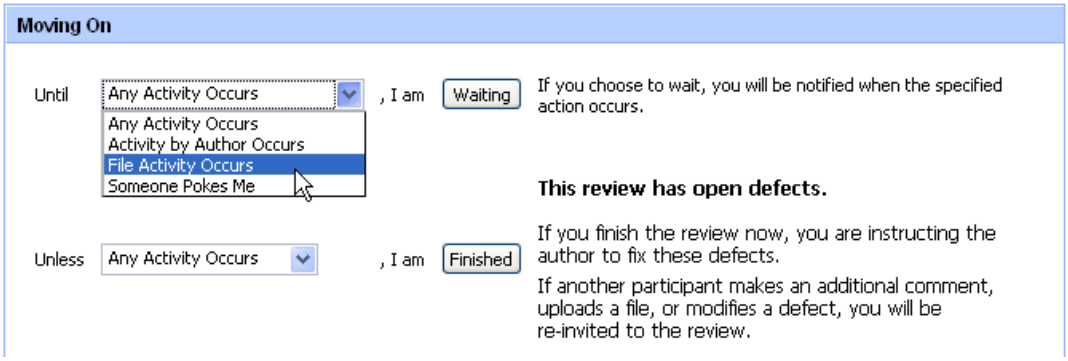
- Defects – When you find a defect, click on the line (in text files) or area (in supporting docs) to which you want to attribute the defect. Make sure the *Add Defect* tab is selected and write a description of your defect, filling in the fields underneath. Once a defect is created, you can mark the defect as fixed, edit or delete the defect, or make it an external defect by using the links at the bottom of the comment box. If defects are entered, authors should fix the defects and upload the fixes, and reviewers should inspect the file again until the content is acceptable. Note that a review cannot be completed until all defects are resolved: either marked as fixed, deleted, or tracked externally 442.

7. Finish Review

The Moving On section at the bottom of the Review Summary Screen allows you to indicate whether you are finished with the review or waiting on further actions from other participants. Both option buttons will have drop-down menus to the left with the same action options: *Any*

© 2003-2011 SmartBear Software

Activity Occurs, *Activity by Author Occurs*, *File Activity Occurs*, and *Somebody Pokes Me*.


Moving On
Until
Any Activity Occurs
, I am
Waiting
If you choose to wait, you will be notified when the specified action occurs.
Any Activity Occurs
Activity by Author Occurs
File Activity Occurs
Someone Pokes Me
This review has open defects.
Unless
Any Activity Occurs
, I am
Finished
If you finish the review now, you are instructing the author to fix these defects.
If another participant makes an additional comment, uploads a file, or modifies a defect, you will be re-invited to the review.

If the selected activity occurs when you are waiting or finished, PeerReview Complete sends you a notification and re-invites you to the review.

- Waiting – *Waiting* keeps you deactivated from the review *until* the action option you have selected occurs. This option signals other participants that you have temporarily stepped out from the review but plan to return.
- Finished – Selecting *Finished* signals to other participants that you are done with the review *unless* the selected activity occurs. If you don't return and the selected activity does not occur, the review remains marked finished by you.

1.4 How Do I?

This section is included as a quick help to get you introduced to some basic tasks in PeerReview Complete. This section is formatted in a series of questions and answers focused on describing how to use PeerReview Complete. While the rest of the manual will serve as the ultimate reference manual, explaining each part or component of PeerReview Complete in detail, this section will be useful if you are new to PeerReview Complete and want to learn how to perform tasks in a quick and easy way. The following questions may not include all your questions, so if you find yourself in need of help, please contact our support team whose information is listed in the Contact Us 24 section.

1.4.1 How Do I Get Started?

How do I...

See a typical review in action? 17

Get started with PeerReview Complete? 17

Experiment with PeerReview Complete without installing the product? 16

Make sure reviews are worth the effort without taking too much time out of my work day to try them out? 17

Perform my first review? 17

© 2003-2011 SmartBear Software

See a typical review in action?
To get introduced to PeerReview Complete, we recommend viewing a demo to see how it works and how a typical review is done.

You can also attend a webinar (either live or recorded), where we go through a more in-depth demo and also answer your individual questions in real time!

Get started with PeerReview Complete?
If you're new to PeerReview Complete, we have heaps of reference materials that help you use and understand the product.

If you want to start evaluating PeerReview Complete or get a quick start with the basics, we suggest the Getting Started guides. Getting Started with PeerReview Complete Server helps system administrators install and configure the main PeerReview Complete server, while Getting Started with the PeerReview Complete Client helps developers get started conducting reviews with PeerReview Complete.

System Administrators 3

Developers 10

We also have the User's manual (a.k.a. Owner's Guide) that serves as a reference manual for everything you want to know about PeerReview Complete. It's divided into sections for installation and server administration, web reviews and the web browser client, the specific integrations, and external integrations.

And if you still have questions, you can always contact us 24. We'd be happy to assist you with any specific questions about PeerReview Complete.

Make sure reviews are worth the effort without taking too much time out of my work day to try them out?

We think reviews are awesome, but we understand you might be skeptical. And why invest in something you're not sure of? We put together some tips on how to minimize your time evaluating the peer review process. This document, Is Code Review for You?, explains how to try peer review for a week, how to see if it delivers solid, measurable results for your team, and how to get your team on board.

Perform my first review?
So you've been assigned your first review, and you're not sure what the next steps are. We'll guide you through so you can jump right in. Just read our blog post here.

1.4.2 How Do I Do Reviews?

How do I...

Know what parts I really need? What are all these parts? 18

Create a new review? 19

Get around having to *Read* all comments? 19

Figure out what "Accept" means? 20

Know why (or when) was (or will I be) I brought back to a review? 20

© 2003-2011 SmartBear Software

Put a review on pause? 20

Finish a review/Clear an action item? 20

Finish a stalled review? Perhaps someone left for vacation, and now the review cannot be completed. 20

Indicate file review order? 20

Remove a participant from a review? 20

Change my notification level? 21

Remove a file? 21

Compare files or choose which version of a file to look at? 21

Change the diff viewer? I don't like the diff viewer. 21

Find out how much time I have spent on each review? 21

Add files to a review from Visual Studio? 21

Review Office Documents? 21

Delete/redact a comment? 21

Compare ClearCase files that are on two different branches? 22

Know what parts I really need? What are all these parts?

Every organization needs a PeerReview Complete Server installed and set up. (Read the Getting Started for Server Administrators 3 for instructions.) The server process acts as the hub, manager, and controller of information. (Download the Server Installer for this component.)

To connect to the server, you use the Web Browser Client 124 by entering a URL (provided by your PeerReview Complete server administrator) into your web browser. The Web Browser Client is where users and administrators can do everything — create and perform reviews, configure personal and system-wide settings and run reports.

The following clients are all optional but very useful. Which one works best for you depends on your version control system, code review environment, and/or personal preferences. Just read the descriptions, and choose what you need or want to use.

The Command Line Client 196 is a cross-platform client interface to the PeerReview Complete server. It can be used for uploading files, integrating with version control, and querying the server, or as a part of an automated script in a sophisticated ALM / build system.

The GUI Client 189 is a cross-platform graphical client to complement the web-based user interface already provided by the server. Use it to upload materials to a new or existing review easily by integrating with any or a combination of the following Version Control Systems: AccuRev, ClearCase, CVS, MKS Source Integrity, Perforce, Rational Synergy, Rational Team Concert, Subversion, Surround, Star Team, Team Foundation Server, and Vault. (If you use a different VCS, you can always use the Command Line Client.)

© 2003-2011 SmartBear Software

The Tray Notifier 278 is a taskbar icon that shows whether you have any pending tasks in PeerReview Complete. This icon allows you to easily access pertinent reviews and alerts you of any new activity in reviews without requiring active involvement in the web browser client.

The Perforce P4V/P4Win plug-ins 381 integrate with the Perforce GUI clients and allows you to upload changelists into new or existing reviews just by right-clicking on the changelist.

The Visual Studio Add-in 286 gives you the ability to upload files and create reviews from within Visual Studio.

Create a new review?

In the Web Browser Client 124, creating a new review is as simple as clicking on the **New Review** button at the top right of the PeerReview Complete page and filling out the information in the next screen.

Wait, you don't see a **New Review** link? Congratulations, you're the PeerReview Complete administrator. By default, we don't allow admins to participate in reviews, so the link at the top right does not show up, and the **Create a New Review** link under the main menu towards the middle of the screen is disabled and appears as "strike-through" text. But you can change this setting by going to the **Admin** link on the top right of the screen. Once you're at the Admin menu, scroll down to the "Review Process Options" and set the *Allow System Administrator to Perform Reviews* setting to *Yes*.

In the Command Line Client, use the command: `ccollab admin review create` to create a new review.

You can also create a new review when adding review materials from the GUI Client, the Visual Studio plugins, and the P4V plugins.. The client will give the option to add the material to a new review or an existing review. Be sure to select *Create New Review* if you want a new review.

Get around having to *Read* all comments?

In the default configuration, you must *Read* all comments. But you can change this! As the system administrator, you can change this setting; go to the **Admin** menu, look under the sub-category **Roles**, select a Role "set", and change *Required to Read All Comments* to *No*. This setting is configurable for each role, so you can ensure that all comments are read by someone, but not everyone.

When this setting is configured to *Yes*, a review participant cannot click the Finish button if the review contains unread comments. When set to *No* for a role, this setting allows a review participant to click the Finish button even if there are unread comments. Also, you will not be brought back to a review if some other participant has commented after your clicked the Finished button. Read our extended blog entry about this feature here.

But my system administrator won't change this setting...

You'll have to mark all comments read, but you can do this easily by clicking the super **Mark All Comments as Read** button at the top left of the Diff Viewer. You'll still have to click it for every file (hey, we have to make sure you're looking at the files) but it saves you clicks in the end.

© 2003-2011 SmartBear Software

Figure out what "Accept" means?

We get this question often, and the short answer is: "Accept" means whatever you want it to mean, so you don't even have to use it if you don't want to. "Accept" does not have an explicit meaning in PeerReview Complete; it is used differently in different environments. So, you can use this feature however you think best suits your review environment. Once this button is clicked, a green checkmark appears to notify participants that a comment has been Accepted." (For more information on the Accept button, read our blog entry here.)

Know why (or when) was (or will I be) I brought back to a review?

You may have been brought back into a review because of your *Required to Read All Comments* setting (See above). Read more about this here.

Put a review on pause?

If you can't continue or finish reviewing because you are waiting for more information (e.g. an answer to a question you posed in a comment), you can put the review on hold by using the **Waiting** button in the Moving On section. "Waiting" keeps you deactivated from the review until the action option you have selected occurs. So the **Waiting** button is for when you want to come back to a review at a later time. If you don't return and the selected activity does not occur, the review will not be marked finished, so to prevent a review from stalling, be sure to come back and finish it.

Finish a review/Clear an action item?

So you want to finish your part of the review. If you're in a role that has the ability to mark reviews finished, it's pretty simple. First, defects must be marked resolved: either *fixed*, *deleted* or *tracked externally*. Then, under the "Moving On" section, select an action option in the drop-down menu to the left of the **Finished** button and click **Finished**. If you are required to read all comments, make sure all comments have been marked read before clicking **Finished**. You'll also be brought back to the review if another comment is made after you have finished a review.

Finish a stalled review? Perhaps someone left for vacation, and now the review cannot be completed.

Several things may prevent a review from being marked *finished*. If an open defect is keeping you from closing a review, have a system administrator resolve the defect, and you'll be able to finish the review. If the absent participant is the only person who has the ability to finish reviews, simply change that participant's role in the review so that they are assigned to a role that is not required to finish the review. For more detailed information, read the blog post, "Ending a Review Now", here.

Indicate file review order?

You can use the "General Chat" and the "Notes" areas to annotate files and give instructions on suggested file review order. For detailed step-by-step instructions, read our blog post here.

Remove a participant from a review?

Removing a participant from a review simply requires editing the participants list in the desired review. Just set the field that contains the participant you wish to remove to a blank setting. Depending on your role configurations, you may have to replace that participant with another person to meet the minimum number of participants required for that role.

© 2003-2011 SmartBear Software

Change my notification level?

You can change the frequency at which you receive notification emails by going to your **Preferences** menu and then setting the *Notification Level* in the **Notifications** tab. When your individual preferences are set to the default setting, *Minimal*, you only get notifications resulting from other users' actions. You can also change the setting to *None* or *All*.

Remove a file?

Once a review has started, removing a file is only possible if there are no comments or defects linked to the file. To remove a file, click the **Upload** button on the "Review Materials" section of the review summary screen. You should see a **[Delete]** link to the left of the file or changelist, in which case click the link to delete the file.

If you do not see the link, it indicates that comments or defects are linked to the file or changelist; you will not be able to delete the file because it would orphan comments.

Compare files or choose which version of a file to look at?

To view different versions of the file you are reviewing in the Diff viewer, click on **Compare** at the top of the Diff viewer. PeerReview Complete shows you a list of all available versions that you can choose from. Use the radio buttons for the right and left side of the Diff Viewer to select the versions. To view only one version, set both left and right sides to the same version.

You can also compare a current version to the latest Accepted version, which is detailed here.

Change the diff viewer? I don't like the diff viewer.

You can't change the aesthetics of our Diff Viewer, but if you prefer other Diff Viewer software, we do support an External Diff Viewer launcher, which you can use to review diffs in a separate diff viewer of your preference. To configure this capability, please visit the owner's manual here 280.

Find out how much time I have spent on each review?

PeerReview Complete v6.0's new user oriented reports allow you to view your own statistics even if you are not an administrator. View our blog post here.

Add files to a review from Visual Studio?

PeerReview Complete v6.0's Visual Studio Add-in offers support for adding files to PeerReview Complete from Visual Studio. To learn more about this add-in and feature, visit our blog here.

Review Office Documents?

Reviewing Microsoft Office Documents is a breeze with our Print To Review that sends the document straight to PeerReview Complete. To view the full steps on how to install and use the printer driver, please view our blog post here.

Delete/redact a comment?

While we still don't allow the deletion of comments, we do provide a way to indicate that a comment is now irrelevant and should not be considered for the review. Detailed steps and screenshots are available here.

© 2003-2011 SmartBear Software

Compare ClearCase files that are on two different branches?

To compare file versions that are on two different branches in ClearCase you have to specify the specific ClearCase version identifiers. Beginning with v6.0 of PeerReview Complete you can use our GUI Client to do that; detailed steps are available here.

1.4.3 How Do I Do Setup or Administrate the Server?

How do I...

Switch from internal authentication to LDAP? 22

Reset my password? 22

Upgrade to a new version of the PeerReview Complete server? 22

Change administrative control (especially in LDAP)? 23

How do I view metrics and reports? 23

Collect data from each review participant? 23

Find metrics for non-text documents? 23

Define groups of users? 23

Switch from internal authentication to LDAP?

The easiest way to change authentication options is to re-run the installer again. There's no need to uninstall, just run the installer over the existing installation. From there, you can choose "LDAP Authentication" when asked and provide your LDAP credentials.

Reset my password?

Sorry, you'll have to ask your PeerReview Complete system administrator about this one.

If you are the administrator and lost your password, view our instructions how to reset your password here.

Upgrade to a new version of the PeerReview Complete server?

1) Back up first! 46

2) Note your Support and Upgrades date by going to **Admin>Licensing**. In the Current Licensing box, take note of the date next to the *Upgrades Expire On:* field. Make sure the version you want to upgrade to was released before this date. You can find this information by looking at the dates next to the version numbers in our Version History 462.

3) Run the new version installer over your existing installation and choose *Existing Configurations* when prompted.

Note: Be sure to fully read our Server Upgrades section in our Owner's Manual here 37.

© 2003-2011 SmartBear Software

Change administrative control (especially in LDAP)?

Administrative Access can be given to individual users by an existing administrator. Just have an administrator go to **Admin>Users** and edit the account you'd like to give administrative control to. On the Editing page, set the *Is Administrator* field to *Yes*. Administrative Access allows a user to access to all reviews, reports, and also edit any review.

The Special Administrator Account is a privileged account that is always allowed to login regardless of how many licenses are in use. With Internal Authentication, the special administrator account is always "admin". This cannot be changed.

With LDAP, there is one user assigned as the "System Administrator". If you would like to change this assignment, go to the ROOT.xml configuration file at *$INSTALLDIR/tomcat/conf/Catalina/localhost/ROOT.xml*. Near the bottom of that file is a line:

<Parameter description="The name of the PeerReview Complete system administrator who is always allowed to log in." name="system-administrator" override="false" value="admin"/>

Change the value to the login name you'd like to set the account to:

<Parameter description="The name of the PeerReview Complete system administrator who is always allowed to log in." name="system-administrator" override="false" value="XXXXXXXXXXXX"/>

where XXXXXXXXXXXX is the login name of the System Administrator. Then restart the service.

How do I view metrics and reports?

PeerReview Complete provides built-in reports that allow you to view detailed metrics by reviews [179], defects [180], and users [180]. To view these reports, you must have access to reports by enabling the Reports Access [76] setting in the **Admin** menu. Once you have access, you may view built-in reports by clicking the **Reports** button that will appear at the top right menu of the PeerReview Complete screen.

Collect data from each review participant?

Participant Custom Fields [107] allow administrators to collect data. To learn about enabling and utilizing this feature, please view our blog post here.

Find metrics for non-text documents?

Additional metrics for binary files, such as Microsoft Office documents and PDFs, can be viewed in v6.0. To view these metrics for a single review, visit the Reports [176] section (provided you have access) and view a Review Detail Report [179]. Our blog post here includes a helpful view of the new metrics to look for.

Many of our customers use our custom reporting [426] features to obtain reports tailored to their needs. Learn how to create custom SQL reports here [435] or learn about our database schema here [427].

Define groups of users?

You can either set up user groups manually or by syncing them automatically to externally defined groups.

To learn how to sync, view our the Group page here [89].

To set up user groups manually, view our owner's manual here [92].

We've also provided some examples of how to use groups that may help you in deciding how to fully utilize this feature here [88].

© 2003-2011 SmartBear Software

1.5 Contact Us

If you have questions or feature requests, drop us a line, or give us a ring.

For Technical Support:
support@smartbear.com

512.257.1569 or 877.501.5651 (M-F, 9am-5pm Central)

For Sales:
sales@smartbear.com

978.236.7900 (M-F, 9am-5pm Eastern)

These links might also be helpful:

Website:
http://smartbear.com

Downloads:
http://downloads.smartbear.com

2 Server Administration

This chapter is useful for system administrators. Regular users of PeerReview Complete should refer to the Web User's Guide 124 instead.

This chapter covers everything from installation 31 / upgrades 37 to backup 46 to licensing 41 to general system configuration 72.

2.1 Installation

2.1.1 Database Installation

PeerReview Complete supports several databases:

- **Embedded** 25. A zero-configuration embedded database useful for trials
- **MySQL** 25. v4.1, v5.0, v5.1.
- **Microsoft SQL Server** 27. 2000 (v8.0), 2005, 2008.
- **Oracle** 29. 10gR2

© 2003-2011 SmartBear Software

For trials, the zero-configuration embedded database is the easiest way to get started. MySQL is the best choice if you need a free, open-source database.

A built-in database migration tool 46 allows you to move between any of the supported databases, so you can always get started on one and switch to another later on.

Note: Oracle and SQL Server are only supported in PeerReview Complete and are not supported by CodeReviewer. For a complete list of differences between the two products, please see the comparison page.

For more about the database format, see the section on custom reporting 426.

2.1.1.1 Zero-Configuration

The server installer comes with support for the Hypersonic embedded database. This in-memory database is perfect for trials where you want to get up-and-running as easily as possible.

Using this database requires no configuration. When the server installer 31 asks for the database type, simply select "Embedded."

While the embedded database is great for quick trials, performance goes rapidly downhill once you have more than a handful of reviews, so for deployment we require a "real" database.

The standard database migration technique 46 works with all databases, so it's possible to move all your data from the embedded database into a "real" database at any time. In addition to using the database migration technique, to move to a "real" database, you will also need to reconfigure the PeerReview Complete server by running the Server Installation program 31.

2.1.1.2 MySQL

MySQL is a popular enterprise-class, open-source database. This is your best choice if you need a free, open-source database.

Downloading MySQL
For downloads and documentation for all platforms, see the MySQL Developer Zone web site. You'll need the MySQL Community Server component for the database itself. You can choose to install the gui tool, `MySQL Workbench`, for graphical server administration, or you can use the MySQL Command Line Client. The MySQL Command Line Client is installed as part of the database server, while the gui client is a separate install that you can get from the developer zone.

Installing the Database Server
First, install the Database Server component. Once you've installed the server, the last screen will give you the option to Configure the MySQL Server. If you select this, it will open a new wizard for configuration. Two parts in the configuration that are especially important: Port Number and Root Password. You will need to know both while getting your database setup and while configuring PeerReview Complete to work with your database. For information on configuring your database server, go to the MySQL Documentation and select your MySQL version.

NOTE: It is important to make sure you are using InnoDB tables. Using InnoDB tables will scale better for multiple users than MyISAM. For questions regarding configuring you MySQL database tables, see http://dev.mysql.com/doc/refman/5.0/en/mysql-config-wizard-database-usage.html.

© 2003-2011 SmartBear Software

Configuring the Database

PeerReview Complete requires that a database be created. The MySQL Server install will not create a database for you. This is a manual step that you need to do.

You can create a database using the MySQL Command Line Client, or using MySQL Workbench. If you plan to use the MySQL gui client, it is important to know that the MySQL GUI Tools Bundle is reaching end of life. The instructions given in this documentation use the functionality in the current gui client, MySQL Workbench.

The steps below for creating a database were written using MySQL Workbench 5.2.16 and are subject to change. As always, the most reliable source for database creation steps is MySQL Workbench Documentation.

1) Open MySQL Workbench.
2) Create a connection to your database server.
   a) From the Home screen, under Server Administration, select New Server Instance.
   b) Follow the steps in the New Server Instance Wizard.
3) Create the database.
   a) Go back to the Home tab.
   b) Under SQL Development, select the database server connection you provided in step 2.
   c) In the Object Explorer, right-click and select Create Schema.
   d) A screen will appear and prompt you to provide a schema name. Once you've done that, click Apply.
   e) You will be shown the command used to create the schema, and given an option to edit it.
   f) Once the create command is as you want it, click Apply.
   g) You will be returned to the screen you saw in step 3d. Click Finish.

We recommend that you create a user specifically for your PeerReview Complete database rather than using the super-user, root. To do this in MySQL Workbench, see the documentation at MySQL Schema Privileges.

All of these changes go into effect immediately. You do not have to restart the MySQL server for changes to take effect.

During the GUI installation screens 31 for the PeerReview Complete server, you will be prompted for the MySQL server host name, TCP/IP port, database name, user name, and password. The installer will report any connectivity errors. When you visit the web page for PeerReview Complete it will detect that you have a new database and will create all tables, indexes, and views for you automatically.

The following SQL script can be used to create a database and a database user for PeerReview Complete and configure the required permissions for that user (be sure to change the database name, user login and password as appropriate):

```
CREATE DATABASE IF NOT EXISTS ccollabdb CHARACTER SET utf8 COLLATE utf8_unicode_ci;
CREATE USER 'collabuser' IDENTIFIED BY 'password';
GRANT
```

© 2003-2011 SmartBear Software

```
   ALTER,
   CREATE,
   CREATE TEMPORARY TABLES,
   CREATE VIEW,
   DELETE,
   DROP,
   EXECUTE,
   INDEX,
   INSERT,
   LOCK TABLES,
   SELECT,
   UPDATE
ON TABLE ccollabdb.*
TO collabuser;
FLUSH PRIVILEGES;
```

MySQL Limitations

1. MySQL v4.x does not support the concept of a database VIEW object. Therefore custom reporting views [432] are only present with v5.x servers.
2. **Important:** The MySQL native restore feature does not automatically clear the database before restoring from backup. You can use the MySQL native restore feature, but you **must** manually drop all tables in the database before you run the restore. By doing this, you'll ensure an exact reproduction of your backed-up database. Option two is to use the alternative method [46] provided by PeerReview Complete.

2.1.1.3 SQL Server

This manual assumes you already have Microsoft SQL Server 2000 (v8.0), 2005, or 2008 installed and running.

Note: SQL Server is only supported in PeerReview Complete and CodeCollaborator and is not supported by CodeReviewer. For a complete list of differences between the two products, please see the comparison page.

Instructions for Microsoft SQL Server 2008 and alternate JDBC drivers

There are a number of different JDBC drivers that can be used for SQL Server - no single driver supports all the versions of SQL Server that PeerReview Complete supports. You should refer to the SQL Server Developer Center documentation for specific details about your database installation.

We ship with a driver that is compatible with many configurations, but not SQL Server 2008. The following drivers can be installed and used in place of the shipped driver:

This driver supports SQL Server 2000, 2005 and 2008 running Java 1.4 or later:

Microsoft SQL Server JDBC Driver 1.2 Download

This driver supports SQL Server 2000, 2005 and 2008 running Java 5.0 or later:

Microsoft SQL Server JDBC Driver 2.0 EULA and Download

To install one of these drivers, or other JDBC driver:

1. Download the preferred driver.

© 2003-2011 SmartBear Software

2. Stop the PeerReview Complete server, if it is running.
3. Go to the following location: PeerReview Complete installation directory/tomcat/lib
4. Replace the `sqljdbc.jar` in that directory with one just downloaded.
5. Start the PeerReview Complete server.

SQL Server 2008 running with Java 6.0
There are additional JDBC driver component files required for SQL Server 2008 support on Java 6.0, and for backward compatibility issues. The Microsoft 2.0 driver referenced above provides a different .jar file for Java 6.0 installations. The driver contains two .jar files: sqljdbc.jar and sqljdbc4.jar. Only sqljdbc4.jar will work on Java 6.0. To install the driver, delete any existing sqljdbc.jar file, and copy the sqljdbc4.jar into the lib directory. Please refer to the SQL Server 2008 JDBC driver documentation for details:

`JDBC Driver Requirements on SQL Server 2008`

Setting up the database
Create a database for PeerReview Complete. It is recommended that you also create a username/password pair just for PeerReview Complete and give this account full access to the database and no access to other databases. You should use a SQL Server account instead of Windows-based authentication because the PeerReview Complete service might not be running under a normal Windows-based login.

During the GUI installation screens[31] for the PeerReview Complete server, you will be prompted for the SQL Server host name, TCP/IP port, database name, user name, and password. The installer will report any connectivity errors. When you visit the web page for PeerReview Complete, it will detect that you have a new database and will create all tables, indexes, and views for you automatically.

SQL Server using integrated authentication

If you are using integrated authentication, there are a couple of additional steps you may need to follow to get PeerReview Complete talking to your database.
1) After downloading the necessary drivers from the links above, extract the files.
2) Copy the platform-specific sqljdbc_auth.dll' to the tomcat subdirectory of your PeerReview Complete install.
3) Go to '.../tomcat/conf/Catalina/localhost/ROOT.xml', and add the following to the end of the JDBC "url":';IntegratedSecurity=true;'
4) Restart the PeerReview Complete service.

SQL Server limitations
With SQL Server 2000, text fields can contain a maximum 8000 characters. This includes changelist descriptions, review overview text, and comment and defect text. With SQL Server 2005 and later there is no such restriction.

© 2003-2011 SmartBear Software

Troubleshooting

1) If you get the error "The TCP/IP connection to the host has failed. java.net. UnknownHostException", you probably need to enable TCP/IP for your database. By default SQL Server has the TCP/IP protocol disabled. PeerReview Complete needs this option enabled in order to access your database. Open the SQL Server Configuration Manager and navigate to "Network Configuration" -> "Protocols". Make sure TCP/IP is enabled. By default, SQL Server is configured to use dynamic ports, which means that the port used is changed each time the service is restarted. To use static ports instead open properties for "TCP/IP" and change the "IPAll" settings. Set "TCP Dynamic Ports" to blank and "TCP Port" to the port number you want to use.

2) By default, when PeerReview Complete is installed, the service runs as the Local System account. This means, if you are using integrated authentication, the Local System account must have the permissions necessary to communicate with the database. Seeing an error such as "Error: Login failed for user 'somedomain\someuser$'." may imply that the local system does not have those permissions. To handle this situation, you can either grant your Local System account permissions to your database server, or edit the account the PeerReview Complete service is using. To edit the account:

   a) From the Administrative Tools menu, open Services.

   b) Find peerreview-server in the service list and stop the service.

   c) Right-click on peerreview-server, and choose Properties.

   d) Go to the Log On tab.

   e) Select the "This account" radio button and provide the account information with which you want the PeerReview Complete service to operate.

   f) Click Apply.

   g) Start the peerreview-server service.

3) If you are using SQL Server with named instances, it is important to know your port configuration. By default, named instances use dynamic ports. This means, every time SQL Server is restarted, it will search for available ports and assign one to your database. In this situation, you might have trouble finding the port upon which your database is running and could see errors when trying to connect to your database with PeerReview Complete

4) . If you are using named instances with dynamic ports, do not specify a port when providing PeerReview Complete your SQL Server connection information. By leaving the port unspecified, a request will be sent to your SQL Server instance on port 1434 that will search for the port your database is running on and will then send that information back to PeerReview Complete.

2.1.1.4 Oracle

This manual assumes you already have Oracle Database Server installed and running.

Note: Oracle is only supported in PeerReview Complete and CodeCollaborator and is not supported by CodeReviewer. For a complete list of differences between the two products, please see the comparison page.

Supported Versions

The versions we've tested against are:

© 2003-2011 SmartBear Software

- 10gR2 Express Edition
- 11gR2 Personal Edition

Your mileage may vary with other versions. Contact us if you have problems -- we *are* interested in fixing bugs!

Setting up the database

Create a database for PeerReview Complete. It is recommended that you also create a username/password pair just for PeerReview Complete and give this account full access to the database and no access to other databases.

During the GUI installation screens 31 for the PeerReview Complete server, you will be prompted for the Oracle server host name, TCP/IP port (default is 1521), database service name (not the SID!), user name, and password. The installer will *not* report any connectivity errors.

> **Warning:** The database service name is not the same thing as the SID! This change was made by Oracle in version 9iR2.
>
> The database service name is fully-qualified, corresponding to `GLOBAL_DBNAME` in an `.ora` file. An example would be `mysid.mydomain.com`. The database service name is also sometimes referred to as "TNS alias" or "connect descriptor."
>
> The SID is the shorter name, corresponding to `SID_NAME` in an `.ora` file.

When you first visit the web page for PeerReview Complete, it will detect that you have a new database and will create all tables, indexes, and views for you automatically, or give you an appropriate error message if there's a connectivity problem. Connectivity problems should be resolved by re-running the installer.

Oracle Limitations

The length of long strings is limited to 1023 characters. This applies to fields such as custom fields, changelist text, comment text, and defect text.

Troubleshooting

We use the Oracle JDBC driver to connect to your Oracle database. The driver has a few undocumented behaviors that may come as a surprise. There are threads on Oracle's tech support forums about this.

Most of the problems arise in the `GLOBAL_DBNAME` field in your `SID_DESC` entry from your `listener.ora` file. A typical entry might look like this:

```
(SID_DESC =
  (GLOBAL_DBNAME = mysid.mydomain.com)
  (ORACLE_HOME = /appl/oracle1/product/10.2.0.1)
  (SID_NAME = mysid)
)
```

© 2003-2011 SmartBear Software

Most other Oracle-based programs use the `SID_NAME` field to identify the database, but the JDBC driver uses `GLOBAL_DBNAME`. This would cause a connection error in the example above.

Also note that *the database service name is not the same thing as the SID*. This change was made by Oracle in version 9iR2. The installer asks for the database service name, not the SID. The database service name is also sometimes referred to as "TNS alias" or "connect descriptor."

Typically, this means you should use the `GLOBAL_DBNAME` in the installer (i.e. `mysid.mydomain.com`) and not just the SID (i.e. `mysid`).

2.1.2 Server Installation

PeerReview Complete uses a stand-alone web server for a cross-platform, no-client, firewall-friendly user interface.

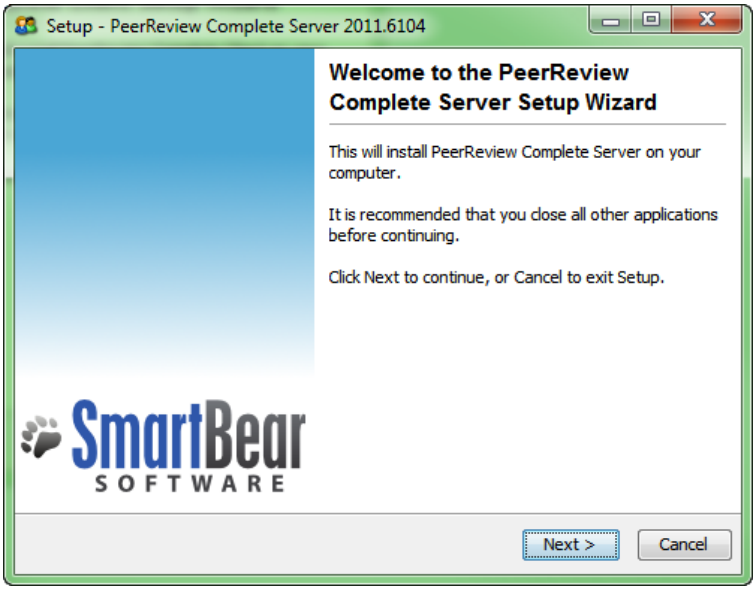
The Graphical Installer

Installers can be run in a graphical, interactive mode or an automatic mode. For first installation, you will need to run the graphical version so you can select options.

First, you need to download the correct installer for your platform from our website:

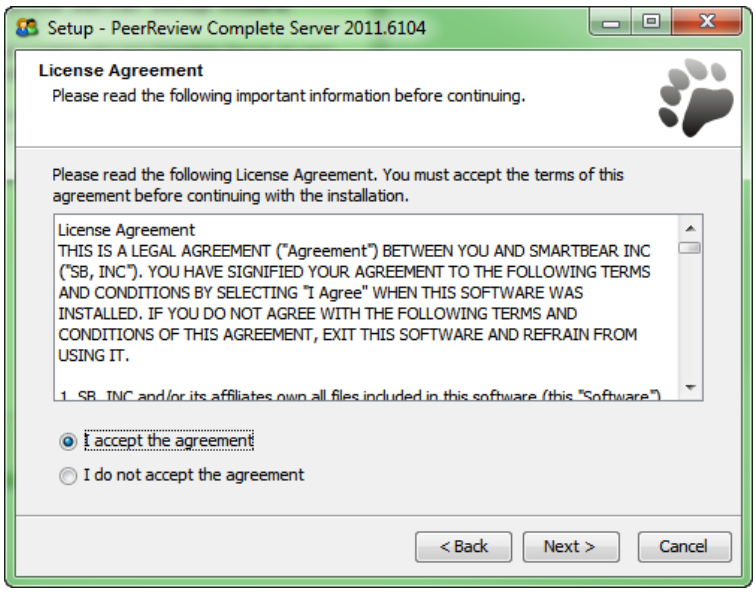
http://downloads.smartbear.com

The first screen of the wizard lets you know it's working:

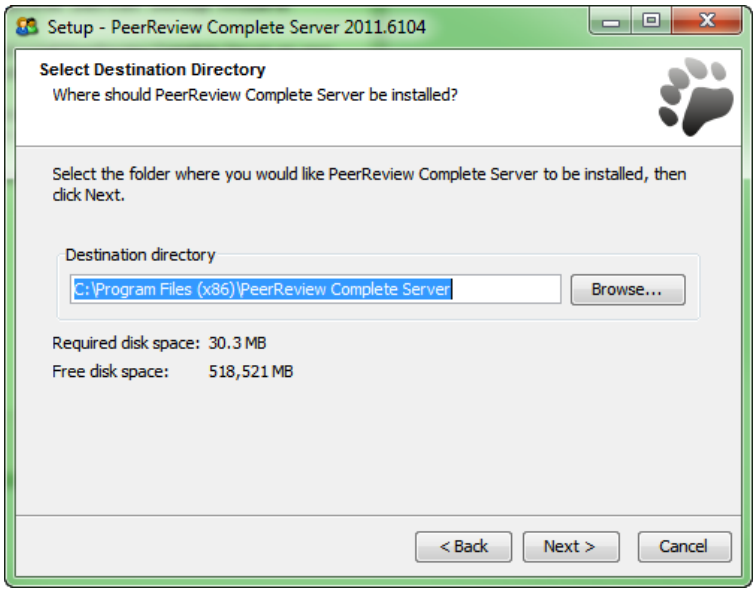

Setup - PeerReview Complete Server 2011.6104
Welcome to the PeerReview Complete Server Setup Wizard
This will install PeerReview Complete Server on your computer.
It is recommended that you close all other applications before continuing.
Click Next to continue, or Cancel to exit Setup.
SmartBear SOFTWARE
Next >
Cancel

© 2003-2011 SmartBear Software

The second screen is the EULA (End User License Agreement):


Setup - PeerReview Complete Server 2011.6104
License Agreement
Please read the following important information before continuing.
Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.
License Agreement
THIS IS A LEGAL AGREEMENT ("Agreement") BETWEEN YOU AND SMARTBEAR INC ("SB, INC"). YOU HAVE SIGNIFIED YOUR AGREEMENT TO THE FOLLOWING TERMS AND CONDITIONS BY SELECTING "I Agree" WHEN THIS SOFTWARE WAS INSTALLED. IF YOU DO NOT AGREE WITH THE FOLLOWING TERMS AND CONDITIONS OF THIS AGREEMENT, EXIT THIS SOFTWARE AND REFRAIN FROM USING IT.
I accept the agreement
I do not accept the agreement
< Back
Next >
Cancel

The next screen allows you to select the disk location for the installation files. Permissions here are important:

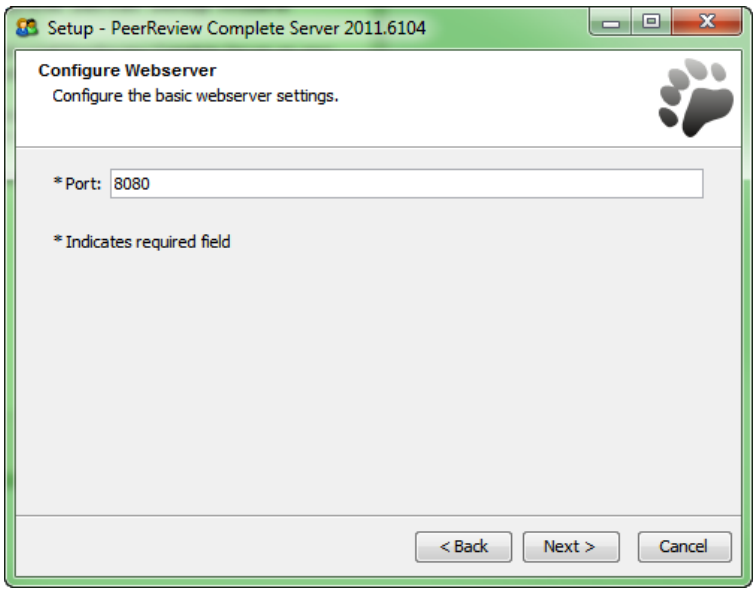

Setup - PeerReview Complete Server 2011.6104
Select Destination Directory
Where should PeerReview Complete Server be installed?
Select the folder where you would like PeerReview Complete Server to be installed, then click Next.
Destination directory
C:\Program Files (x86)\PeerReview Complete Server
Browse...
Required disk space: 30.3 MB
Free disk space: 518,521 MB
< Back
Next >
Cancel

This directory should be writable by the server because several dynamic files are created here including web server logs, temporary storage, and long-term storage for file content uploaded by users. All of these locations can be changed if it is critical that the installation location be read-only, but this requires significant work on the part of the administrator and makes upgrades more difficult.

Make sure at least 5 gigabytes of space is available in the named directory. Your users will need the space for file uploads. You can always move and/or archive 122 the file upload directory.

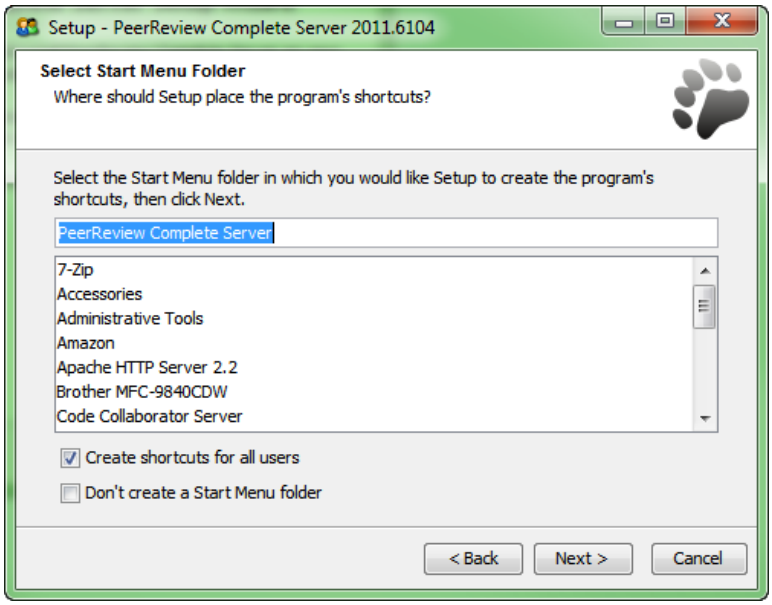
© 2003-2011 SmartBear Software

The next screen configures the web server settings:



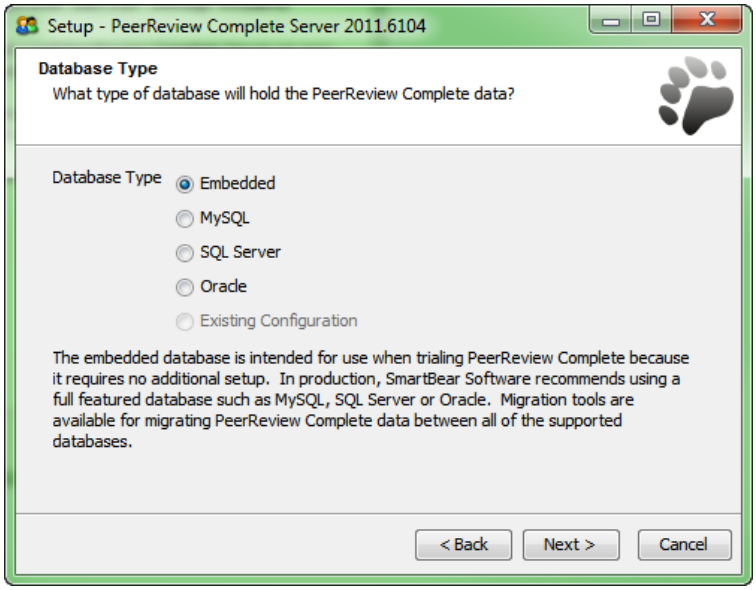
The port number should be selected as to not conflict with any existing services. The installer will attempt to connect to this port when you click `Next`; if the port is already taken by another process, you will get a warning message. You can elect to try a different port or continue. *Continue at your own risk* -- probably the server will not be able to start up and you'll have to make the port available and restart the server manually.

The next screen will ask where you would like to create the PeerReview Complete's shortcut. Click "Don't create a Start Menu folder" if you would not like to create a folder for PeerReview Complete.


Setup - PeerReview Complete Server 2011.6104
Select Start Menu Folder
Where should Setup place the program's shortcuts?
Select the Start Menu folder in which you would like Setup to create the program's shortcuts, then click Next.
PeerReview Complete Server
7-Zip
Accessories
Administrative Tools
Amazon
Apache HTTP Server 2.2
Brother MFC-9840CDW
Code Collaborator Server
Create shortcuts for all users
Don't create a Start Menu folder
< Back
Next >
Cancel

Next, you configure the database connection. The database should already be installed and ready 24:

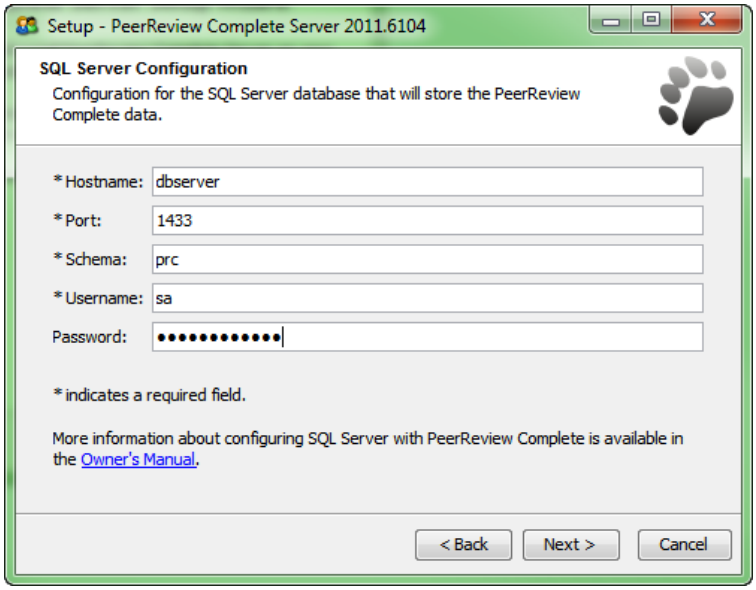
© 2003-2011 SmartBear Software


Setup - PeerReview Complete Server 2011.6104
Database Type
What type of database will hold the PeerReview Complete data?
Database Type
Embedded
MySQL
SQL Server
Oracle
Existing Configuration
The embedded database is intended for use when trialing PeerReview Complete because it requires no additional setup. In production, SmartBear Software recommends using a full featured database such as MySQL, SQL Server or Oracle. Migration tools are available for migrating PeerReview Complete data between all of the supported databases.
< Back
Next >
Cancel

For trial installations, you'll probably want to select the default "Embedded" database [25]. You should migrate [46] to a full-featured database before you use the server in production.

If upgrading servers [37], you should select "Existing Configuration" to choose to maintain the same configuration as your older installation. "Existing Configuration" will only be grayed out during fresh installations where the server has not been previously installed.

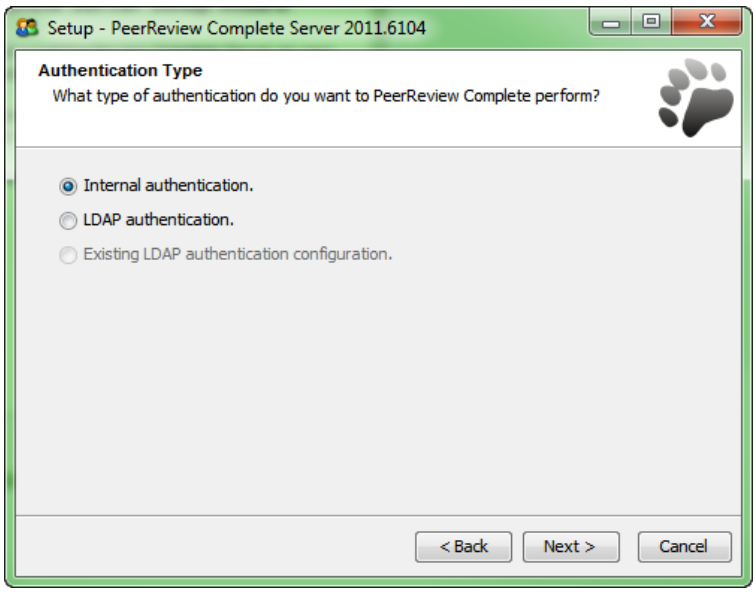
If you picked anything but the "Embedded" database, the next screen lets you configure the connection to the database server:


Setup - PeerReview Complete Server 2011.6104
SQL Server Configuration
Configuration for the SQL Server database that will store the PeerReview Complete data.
* Hostname: dbserver
* Port: 1433
* Schema: prc
* Username: sa
Password: ••••••••••••
* indicates a required field.
More information about configuring SQL Server with PeerReview Complete is available in the Owner's Manual.
< Back
Next >
Cancel

© 2003-2011 SmartBear Software

The exact format of this screen depends on the database you chose. The `Username` and `Password` fields here refer to the *database* username and password. Typically, you will want to create a special database username and password for the PeerReview Complete application so you can control exactly which data it has access to. This usually means full access to the database created for PeerReview Complete and no access to any other database.

The next screen lets you decide how users will be authenticated in PeerReview Complete.


Setup - PeerReview Complete Server 2011.6104
Authentication Type
What type of authentication do you want to PeerReview Complete perform?
Internal authentication.
LDAP authentication.
Existing LDAP authentication configuration.
< Back
Next >
Cancel

The simplest setting is "Internal," which means PeerReview Complete should maintain usernames/passwords in its own database.

If your company uses LDAP or ActiveDirectory, you can use that method to authenticate users in PeerReview Complete, which means you don't have to maintain the PeerReview Complete user list at all. If you select the LDAP method you will get an additional wizard screen that lets you supply the settings for your LDAP server. See the LDAP section 50 for details.

You will also have the option to choose "Existing LDAP authentication configuration" if you are upgrading 37 an existing installation.

© 2003-2011 SmartBear Software

After this, the installer lays down the server files, stopping any existing server if necessary. Once the new files are installed, the server is started automatically, and the installer tells you it is finished:

Setup - PeerReview Complete Server 2011.6104
Completing the PeerReview Complete Server Setup Wizard
Setup has finished installing PeerReview Complete Server on your computer. The application may be launched by selecting the installed icons.
Click Finish to exit Setup.
Thank you for choosing SmartBear Software!
SmartBear SOFTWARE
Finish

After you click `Finish,` a web browser will open, pointed at your installed server.

> **Warning:** If the server hasn't quite had enough time to fully start up, it might take a while for the web page to load and it might even fail to open. Just "refresh" the browser.

The Non-Graphical Installer

If you are upgrading the server instead of installing from scratch, you have the option of installing without any graphical user interface.

We recommend that you always use the graphical interface when possible because it gives you the chance to review settings, especially new settings that we might have added since your last installation.

To run the installer without a GUI, run the installer from a command-line using the `-q` switch. To set the installation directory from the command-line, use the `-dir [directory]` switch. The -q switch gives you a silent install and will not prompt you for any installation instructions. If you'd like to be prompted for installation instructions without using the graphical installer, you can run the installer from the console using the -c command.

Installing more than one server instance on a single machine

All server instances need to be running on unique and available ports. Each server instance will also need a separate database and license code.

On Windows, install the different instances in different directories, on different ports, with separate database instances. This will get all of the software installed, but only the last one will be properly installed as a service. To install the others as a service, run the following command from the command line:

© 2003-2011 SmartBear Software

<install-dir>\peerreview-server /install service-name

This will install that instance as a Windows service with the specified service name. To uninstall a service that was misnamed or no longer used:

<install-dir>\peerreview-server /uninstall service-name

On *nix platforms, you should install the server in multiple directories and then edit your /etc/init.d scripts accordingly.

2.1.3 Server Upgrades

Back Up Before Upgrading!
As with all software, you should always perform a complete backup 46 before attempting an upgrade.

For major releases, we also recommend that you install the new version on a test machine using a copy of your current database. Test a typical workflow on the system to make sure it works and there are no surprises for your users.

Upgrading from Installers
Upgrading the server component works exactly like installing 31 the component.

> **Warning:** Do *not* uninstall the server first. Just install the upgrade over the existing installation.

Warning: You cannot upgrade by skipping versions (e.g. directly from v2.0 to v4.0). You must upgrade from v2.0 to v2.1, then from v2.1 to v4.0.
The installation screens are the same except that existing configuration will be automatically pre-loaded. This means that clicking `[Next]` repeatedly will do the right thing in most cases. Of course, we recommend that you read each screen in case we added additional options.

Windows-Only: The system will be taken off-line while the upgrade takes place and will start automatically after the upgrade completes. When upgrading using a 64 bit installer on a server installed using the 32 bit installer, be sure to verify the install directory is the same as you currently have. The 64 bit installer will NOT automatically detect the existing install directory of a PeerReview Complete install that was done using the 32 bit installer. In this scenario, simply edit the install location to be the fully qualified path of the existing install directory.

Unix-Only: The installer will *not* automatically stop or start the server, so if the server is currently running you will need to manually stop it, perform the upgrade, and then restart it.

Upgrading Database Schema

With minor and major releases, we sometimes change the database schema.

In this case, the database schema needs to be upgraded before the server can be used. After the installer completes and you go to the main web page, you will get a special message saying the database needs upgrading. A button allows you to start the upgrade. Schema upgrades are automatic.

Once the database schema is upgraded, you cannot then go back to a previous version. This is one of the many reasons why backups are important!

© 2003-2011 SmartBear Software

2.1.3.1 Testing Newer Versions

If you are currently using PeerReview Complete and want to try a newer version without risking your existing installation, there are a couple ways to install a newer version in an existing PeerReview Complete environment while still maintaining the integrity of your existing version.

Fresh Installation
This option allows a quick and easy way to try PeerReview Complete from scratch. However, you will not have access to all your existing review and user data.

To test a new PeerReview Complete server with a blank database, download and install 31 the newer version of the PeerReview Complete server software. If you are installing onto the same machine as your existing PeerReview Complete server software, *be sure to specify a different directory, port number, and database instance.* Installing it in the same directory will upgrade your current installation, and you will not be able to "undo".

Parallel Operation
This option allows you to continue running the old version of PeerReview Complete in parallel with the newer version you are testing..

1. Do a complete backup 46 of your existing PeerReview Complete server
2. After the backup finishes, install the exact same build of the PeerReview Complete server software that you are currently using. For example, if you are currently running v5.0.5029, then install build 5029 of v5.0 (older versions of the installer are available here).

   After the installer finishes it will display a web page in your default browser. That web page will have a button for creating the PeerReview Complete database tables - do not push that button! You don't want those tables to be created because you are about to restore from your existing database.

   Again, if you are installing onto the same machine as your existing PeerReview Complete server software, *be sure to specify a different directory, port number, and database instance.* Installing it in the same directory will upgrade your current installation, and you will not be able to "undo".
3. Restore the backup 46 that you just created to the new installation of PeerReview Complete.
4. Then run the installer 31 for the newer version and when prompted, select the directory that contains the PeerReview Complete server that you just installed. This will cause the installer to upgrade that installation.

Windows Notes
If you choose either of the above options, you will end up with two instances of the PeerReview Complete server running: one for your older version and one that is running the newer version. If you are running both instances on the same machine and if the operating system on that machine is Windows, then there is one additional step.

The PeerReview Complete server installer for Windows always uses the same Windows service name: peerreview-server. This means that after you install a second instance of the PeerReview Complete server on a Windows system, the Windows service named peerreview-server points to the new installation. The original installation no longer has a Windows service entry and is therefore no longer running.

© 2003-2011 SmartBear Software

This problem is easy to fix. In a command window set the working directory to the original installation of the PeerReview Complete server software. Then enter this command:

```
peerreview-server /install <service-name>
```

You can specify anything you want for <service-name> except for peerreview-server. This will create a new Windows service with its own name for your existing PeerReview Complete server installation.

2.1.4 Network Configuration

Certain network topologies and configurations require specific configuration in the server component.

Server Proxies
PeerReview Complete optionally will connect to a licensing server hosted by SmartBear via standard HTTP protocol in order to validate your license code. If you are installing PeerReview Complete in an environment where outbound HTTP requests are required to use a proxy, you will need to configure PeerReview Complete to use the proxy.

Proxy settings are configured in the *`installation-directory`*`/ccollab-server.vmoptions` file. To enable an HTTP proxy, you will need to add the following lines to that file:

```
-Dhttp.proxyHost=proxy_hostname
-Dhttp.proxyPort=proxy_port
-Dhttp.proxySet=true
```

Many proxies are configured to reject connections to internal URL's via the proxy, as these connections are supposed to be made directly. For this reason, it is also a good idea to configure internal hosts as non-proxied hosts. Specifically, it is important to configure `localhost` (including its resolvable name), the bug tracking system, and the version control server as non-proxied hosts. These URL's are configured in PeerReview Complete for integration purposes and a connection is made to validate the URL's entered in the configuration screens. If the proxy rejects the connections, you may not be able to properly edit those fields. To configure non-proxied hosts, add the following line to the `collab-server.vmoptions` file:

```
-Dhttp.nonProxyHosts="localhost|collabserver|*.mydomain.com"
```

The format for the value is a list of hostnames delimited by `"|"` and using `"*"` for a wildcard.

Some newer firewalls have been known to cause additional problems with HTTP proxying. For instance, some firewalls periodically redirect HTTP requests to a firewall-generated web page requiring the user to log in. These firewalls are specifically designed to limit web access to users; preventing services (such as PeerReview Complete) from accessing web services. Organizations with such firewalls will need to work with their network administrators to exempt PeerReview Complete from this policy or otherwise allow access.

© 2003-2011 SmartBear Software

2.1.5 Platform-Specific Notes

This is additional server-related information that is specific to certain operating systems.

Windows
The web server is installed as a Windows Service. This means the server starts up automatically when the machine boots up, and no user needs to be logged in. The service can be started and stopped manually and even remotely.

The default installation is for the service to start automatically upon system startup using the default service user. All of these settings can be changed by the administrator after installation using Microsoft's standard service configuration control panel.

> **Warning:** If you change the user under which the service runs, make sure the installation directory is still both readable and writable by the new user.

> **Warning:** If PeerReview Complete is located on the same machine as the database, and if the PeerReview Complete service starts up before the database service, PeerReview Complete might fail to start up. The work-around is to restart the PeerReview Complete service, but the fix is to use service dependencies to tell Windows that the PeerReview Complete service is dependant on the database service. This is a standard Windows service feature.

Linux / Solaris
The web service is not automatically installed such that it will run automatically when the system starts up. However this is easy to set up.

The installation directory contains a file `ccollab-server` that can accept the usual start and stop commands. Create a symbolic link to this file from your standard installation directory (e.g. `/etc/rc3.d` or `/etc/rc.local`) to cause the server to start automatically upon system startup.

> **Warning:** If PeerReview Complete is located on the same machine as the database, make sure it starts up *later* in the start-up process than the database server. Otherwise, PeerReview Complete will fail to start up.

© 2003-2011 SmartBear Software

2.2 System Administration

2.2.1 Licensing

SmartBear offers both fixed-seat and floating-seat licensing plans.

Licensing Options and Definitions
Fixed Seats - With a fixed-seat license, you are allowed a fixed number of "named users." A "named user" is a human being (not a machine) that is active in the past 30 days. Therefore, if some users never log in, or if a user leaves the group and doesn't log in again, that user does not count towards the fixed-seat license. Fixed-seat licensing is more appropriate when most users will be using the product daily.

Floating Seats - With a floating-seat license, you are allowed a maximum number of "concurrent users." A user is considered "active" if he/she has used the system in the past hour *and* has not explicitly logged out. Users with any PeerReview Complete page open in a browser will remain logged in regardless of whether they are actively navigating through the site. Floating-seat licensing is more appropriate when you have many users that will use the system only occasionally.

In both cases, a "user" is a human being, not tied to a particular machine or client. This means that if someone uses both the Eclipse client, stand-alone client, and command-line client, plus uses PeerReview Complete on a work machine and laptop, that all counts as one "seat" whether fixed or floating.

When you disable 86 a user, that user immediately does not count against your license usage. That user will not be able to log in, and if the user is in the middle of something the user will be kicked out of the system.

Licensing "Grace" Seats
Although your server is licensed for a certain number of fixed seats, SmartBear provides you with additional "grace" seats. These appear in the licensing administration screen.

"Grace" allows you to run over your license limits temporarily with no disruption in service. For example, if you have 50 fixed licenses and 10 grace licenses, you can have up to 60 users logged into the system concurrently.

After both grace and standard seats are exceeded, users will see error messages about hitting the license limit. This will prevent general use of the product and you will have to contact us for assistance.

Grace is intended for *temporary* overflow for fixed seats. If you are consistently using grace seats over a period of many days, we will ask you to purchase additional seats.

Exceeding the license limit
How does the system behave when you exceed the license limit?

Once you have maxed out both granted and grace seats, either in fixed or floating schemes, the next user who attempts to log into the system is shown an error message and is not allowed in. The message will explain that all licenses are in use and will prompt that user to contact his system administrator. In the case of a floating-seat scheme, the user is shown the list of all currently-active users.

Note that "log in" can either mean "logging in from the front page" or "going back into the system after hours of inactivity." In the latter case, a cookie is used to automatically log the user back in. Normally, this is completely transparent, but if this auto-login will cause the system to go over the licensed limit, the user will see the error message.

© 2003-2011 SmartBear Software

This all applies to users who aren't already active. The users who were already legally using the system continue uninterrupted. It's the users who come in after that point in time who aren't allowed in.

One way to get back under the limit is to disable 86 user accounts. This works for both fixed and floating seat schemes. If you do this the license is made immediately available for another user. If the user you disabled continues to use the system or tries to log in, that user will receive an appropriate error message explaining he has been locked out and to contact his system administrator.

The system administrator 87 is always allowed to log in, even if the license has been exceeded. This allows you to remedy the licensing problem, either by disabling users until you're under the limit, or to install a new license code that has been provided to you. You can also log out a user if e.g. they have gone home and forgotten to close their browser.

Monitoring license usage
There are several ways to keep track of licenses used.

- On the Users Administration 84 screen current usage numbers are displayed. You can also manually disable or logout users if you need to reclaim licenses quickly.
- On the System Status 123 screen you can see historical activity patterns.
- You can use JMX 60 to monitor usage and licensing information.

How many licenses do I need?
It's hard to give guidance because it varies quite a bit.

Generally the best advice is: Just try PeerReview Complete. There is a report on the User Administration 84 page that tells you exactly how many "fixed" and "floating" seats you would be using right now. Use that empirical information to determine which is cheaper.

Typically, a trial doesn't involve everyone, and typically the usage pattern isn't exactly the same as when it will be deployed, so you'll have to estimate. Still, you'll be doing so with some real numbers.

Here are some additional pointers:

- If you have developers in many timezones, floating usually becomes more economical.
- If each person will be inside PeerReview Complete at least once per day (either as author or reviewer), typically fixed seats are cheaper. If less frequently than that, floating.
- If you expect large spikes in usage -- where perhaps everyone is online at once during a review crunch -- you'll want fixed. Otherwise, you have to get enough floating to handle that peak usage.
- Remember that you can change between fixed and floating later, with no financial penalty. So if it turns out you have the wrong model, it's not a big problem.

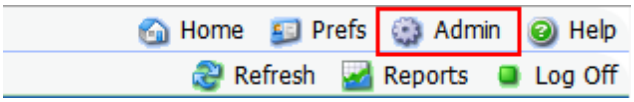
When Requesting Your License Code
In order to generate a license code you'll need to send us the Node ID from your PeerReview Complete server installation.

To find the Node ID, do the following:

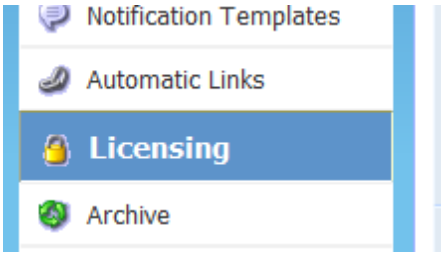
1. Log into PeerReview Complete with any user that has administrative access.

© 2003-2011 SmartBear Software

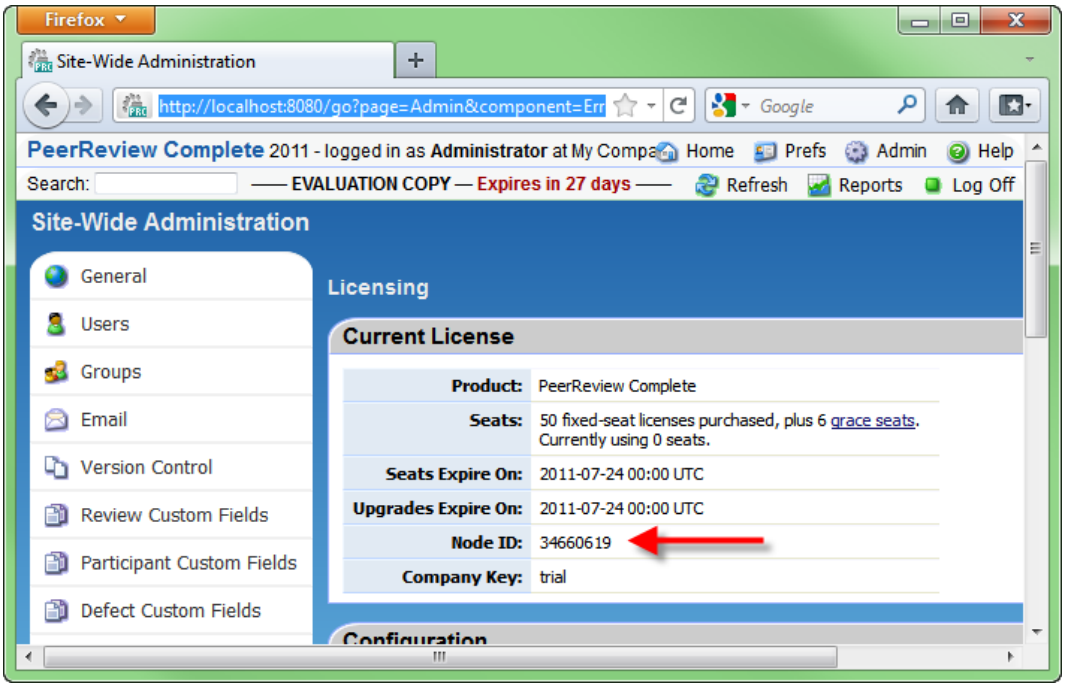
2. Go to the "Admin" screen from the menubar at top-right:



3. Select "Licensing" from the list of sub-categories on the left:



4. There will be a table at the top of the page called `Current License`. Look for the Node ID field, and there will be an 8-character code. That is your Node ID.



Entering Your License Code

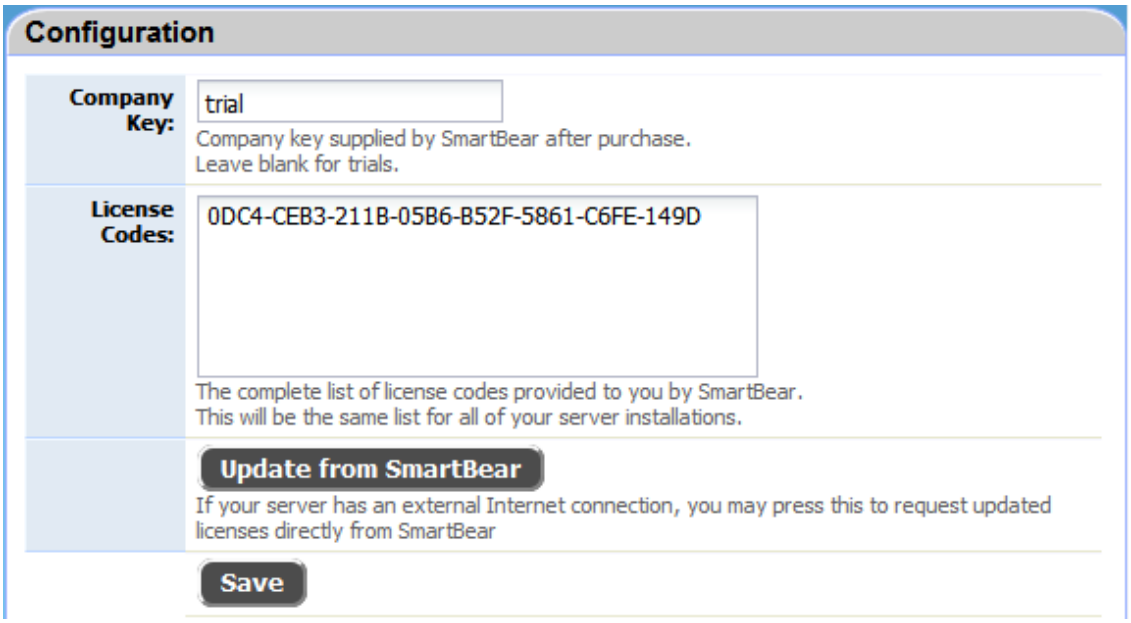
When you receive your license code, the license code is entered into the PeerReview Complete server. Client software does not have to be specially licensed.

To enter your license code, do the following:

1. Go to the `Licensing` page. (See steps 1-3 above as if you were looking for your Node ID 42.)

© 2003-2011 SmartBear Software

2. Below the `Current License` box, there will be a `Configuration` box. In the "Company Key" field, enter the company key. In the "License Codes" field, enter the license code given to you by SmartBear sales:

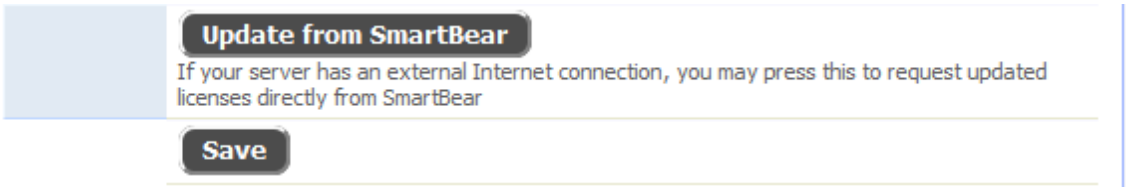

Configuration
Company Key:
trial
Company key supplied by SmartBear after purchase.
Leave blank for trials.
License Codes:
0DC4-CEB3-211B-05B6-B52F-5861-C6FE-149D
The complete list of license codes provided to you by SmartBear.
This will be the same list for all of your server installations.
Update from SmartBear
If your server has an external Internet connection, you may press this to request updated licenses directly from SmartBear
Save

5. Click "Save" to complete changes.
6. If there are any errors they will be displayed at the top of the screen. Otherwise your licensing information (number of fixed and floating seats) will be displayed along with any grace seats.

Updating License Code

You can also receive updated licenses from the SmartBear server after your initial license set-up. You must have an external Internet connection to do so.

Just click on the "Update from SmartBear" button near the bottom of the Configuration box:


Update from SmartBear
If your server has an external Internet connection, you may press this to request updated licenses directly from SmartBear
Save

2.2.2 Content Storage

Most review data is stored in the database. The one exception to this is the contents of the files under review. These are stored in a directory described as the "content cache". The internal structure of the content cache is subject to change between versions of PeerReview Complete, so we do not recommend altering the contents of the content cache directly without specific instructions from SmartBear technical support. However, there are two operations on the content cache that system administrators can perform: choosing a location and archiving old content 122.

© 2003-2011 SmartBear Software

Choosing a Location for the Content Cache

The default location for the content cache is the collaborator-content-cache subdirectory of the install directory. In many environments, this may not be an acceptable location. Here are some things to consider when choosing a content cache location:

- File permissions may not allow writing in the installation directory.
- Maintenance of network storage may be easier if backups are already in place and disk usage is monitored automatically.
- Network storage allows for warm standby PeerReview Complete servers to be available in the event of failure of the primary server.

Configuring the Location of the Content Cache

If a determination is made that the default content cache location is unacceptable, the location can be changed in *`installation-directory`*`/tomcat/conf/Catalina/localhost/ROOT.xml`. Find the `Parameter` element with the name attribute `"content-cache"`. The `"value"` attribute of this element should be the path to the content cache. Relative paths are interpreted relative to the installation directory.

If the content cache location is changed on an active server, it must be restarted before the change will take effect. If the content cache is not also copied over to the new location, users will see a message that content is missing and was probably archived by the administrator. To prevent this, it is best to shut down the server, copy the content cache to the new location, and then restart the server.

Sharing the Content Cache

Sharing the content cache between servers is *not supported*. The only environment where PeerReview Complete servers should be configured to use the same content cache is if one is configured as a warm backup for the primary server. A warm backup is a system that configured, *but not running*. It can be started in the event of primary server failure to reduce down time.

Upgrading Content Store Format

There is a new file content store format that works better with many filesystems by having a deeper directory structure, reducing the number of files in each directory. The format of the cache is determined by the cache.properties file located at the root of the content-cache directory. By default this is *installation-directory*`/tomcat/collaborator-content-cache`. This file is a Java properties file (spec). There are two configuration keys, "version" and "lazy-upgrade-from-version". The "version" key determines whether to use the new format (2), or the old format (1) for storing new data. The "lazy-upgrade-from-version" key determines whether to search for and upgrade data stored in the old format (1), or not (blank). For example, here are the values you would use to store data in the new format and automatically upgrade data in the new format when requested:

```
version=2
lazy-upgrade-from-version=1
```

To just use the new format

```
version=2
lazy-upgrade-from-version=
```

To just use the old format

```
version=1
```

© 2003-2011 SmartBear Software

```
lazy-upgrade-from-version=
```

2.2.3 Backup / Migration

PeerReview Complete can be backed up while it is running. Also, there is a system for migrating data from one database to another.

PeerReview Complete stores almost all data in the database initially set up for it. It also uses a local directory to store copies of uploaded files. These are the two systems that need to participate in the backup/restore process.

Backing up the Database
The mechanism for backing up the database depends on the database. Other than the embedded database that's used for trials only, all supported databases have some mechanism for live backup. See the documentation for your database for details.

There is an alternate method for backing up the database which is to perform the first half of a database migration [47] and save the migration data file.

Backing up the File Cache
The local directory you need to back up is located here (by default -- it can be changed by the system administrator):

installation-directory`/tomcat/collaborator-content-cache`

Typically, a backup mechanism will either copy this directory elsewhere or will keep a zip or other compressed archive file updated with the contents. You can do this while PeerReview Complete is running, although most backup mechanisms will run at off-peak hours.

Files in this cache are stored in such a way that a file is written *only once* and thereafter is never changed. This means incremental backups of the directory are particularly easy -- only new files must be copied. Most file-copy utilities have a mode that means "only copy new files."

Restoring a PeerReview Complete Backup
To restore a PeerReview Complete installation, first restore the database as directed in your database documentation. Then, install the PeerReview Complete server software. Finally, restore the contents of the collaborator-content-cache directory in the new installation directory. You can do this last step while PeerReview Complete is running.

If you've used the database migration technique to back up the database, refer to steps 3-11 of Restoring the Data Dump File [48] for instructions on migrating the database data.

Migrating Data Between Databases
PeerReview Complete has a generic mechanism for migrating data between databases -- even if the databases are completely different types.

Applications of this migration technique include:

- migrating between the embedded database and one of the other databases when a trial moves to a production environment
- backing up a database in a database-independent and easily-inspected manner
- when we add support for a new database and you want to switch over to it

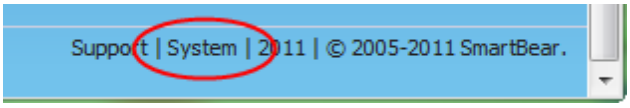
© 2003-2011 SmartBear Software

- we need to debug your database and you need to send a "dump" of your data to SmartBear Technical Support

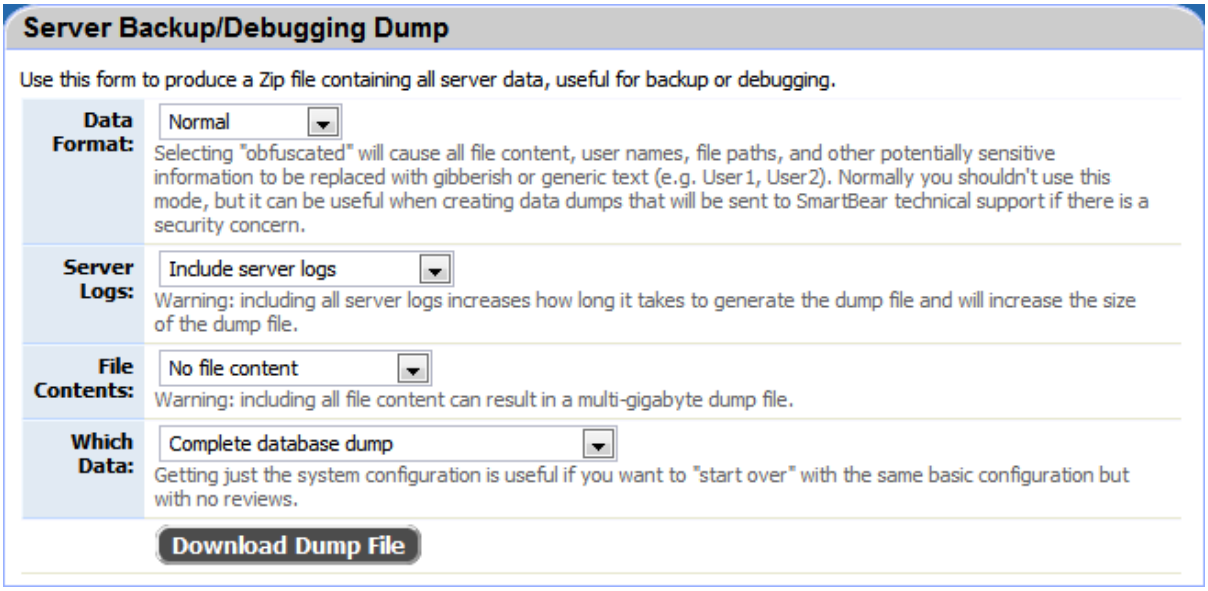
There are two halves to the migration process: Creating the migration data dump file, and "restoring" the data dump file into a PeerReview Complete installation.

1. Creating the Migration Dump File

To create the migration data dump file, log into the PeerReview Complete web server, and click the `System` link at the bottom of the screen:



Near the top of the screen is a form and a button which will allow you to download a whole system dump:


Server Backup/Debugging Dump
Use this form to produce a Zip file containing all server data, useful for backup or debugging.
Data Format:
Normal
Selecting "obfuscated" will cause all file content, user names, file paths, and other potentially sensitive information to be replaced with gibberish or generic text (e.g. User1, User2). Normally you shouldn't use this mode, but it can be useful when creating data dumps that will be sent to SmartBear technical support if there is a security concern.
Server Logs:
Include server logs
Warning: including all server logs increases how long it takes to generate the dump file and will increase the size of the dump file.
File Contents:
No file content
Warning: including all file content can result in a multi-gigabyte dump file.
Which Data:
Complete database dump
Getting just the system configuration is useful if you want to "start over" with the same basic configuration but with no reviews.
Download Dump File

Making the appropriate selection:

1. **Data Format:** Normal - Selecting "Normal" will leave the data as is.

   Obfuscated - Selecting "Obfuscated" will change the data to conceal sensitive information.

2. **Server Logs:** Include server logs - Selecting this will include server logs in dump. Note: If you are creating a dump file to send to technical support, it's important to include the server logs.

   Don't include server logs - Selecting this will not include server logs in the dump file.

© 2003-2011 SmartBear Software

3. **File Contents:** No file content - Selecting "No file content" will not include file content.

Include all file content - Selecting this will include all file content, but this will also cause a longer download.

4. **Which Data:** Complete database dump - This must be selected for a whole system dump.

System configuration only - no review data - Selecting this will only save the system settings.

Once you've filled out the form, click the "Download Dump File" button to download the database dump file. This is a ZIP file containing all your database data in a platform- and database-independent XML format, plus additional files that describes your server environment.

2. Restoring the data dump file

Loading this data into another PeerReview Complete installation requires some effort. This process is intentionally complex to prevent accidental destruction of real data.

Below are the steps to restoring a database migration data dump file:

1. Verify your PeerReview Complete Server install.

   1.1. Make sure the PeerReview Complete server version and build number are identical between the server that created the data dump file and the server that is loading it. If the versions don't match exactly, the restore might not succeed. SmartBear maintains installers for previous versions of the server if you need one. You can always upgrade to another version after the migration.

   1.2. Be sure when running the installer, you give the correct connection parameters to point the server to the database to which you plan to migrate. Note: Even if the PeerReview Complete Server install location is not changing, you will still need to rerun the installer to point PeerReview Complete to the correct database.

   1.3. Once the install is complete, a browser window should open. Do NOT initialize the database as prompted in the browser.

2. Move the dump file into a known location. In this example we'll assume the location is: c:\temp\ccollab-dump.zip.

3. Make sure the PeerReview Complete server where you will load the dump is not running. This will not work on a running server.

4. Open the Tomcat session configuration file located here: installation-directory/tomcat/conf/Catalina/localhost/ROOT.xml

5. Find the parameter called database-migration-data-path, or create one if it doesn't already exist. It should look something like this: <Parameter name="database-migration-data-path" value="c:\temp\ccollab-dump.zip" override="false" />

6. Make sure the value string matches the location of the dump file, as in the example above. Use an absolute file path.

7. Save the configuration file.

© 2003-2011 SmartBear Software

8. Use a database administration tool to make sure the database configured for use with PeerReview Complete has no tables in it. If there are any tables in the database the restore will not work. This prevents accidental restoring over an existing database.

9. Start the PeerReview Complete server.

10. The server will automatically load the data from the database migration data dump file. If there were any problems with migration, you will see a helpful error message in the server log 72. The server will also log progress reports as data is loaded up, so if you have a large database and you wish to monitor migration progress you can "tail" the log file to see what's happening. Loading migration data can take a long time, so be patient!

11. Upon starting after restoring a migrated database, email notifications will be disabled. This is to prevent users from receiving spurious duplicate email notifications when administrators restore into test configurations. You may re-enable email notifications from the email administration page 99.

2.2.4 Security Considerations

PeerReview Complete administrators need to be aware of several security issues and options which affect the overall security of the system. This section covers those issues.

File System Security
PeerReview Complete relies on the underlying operating system as a foundation for overall system security. Several potentially sensitive items are stored in the local filesystem, including database credentials, LDAP credentials (if used), and file contents. Care should be taken to maintain system security of the server's operating system, so this information is not compromised. SmartBear does not have any specific security expertise, so we recommend you follow the guidance of your operating system's vendor.

HTTP Transport Security
By default, the PeerReview Complete server operates over regular HTTP. This means that all communications between clients and servers are unencrypted on the wire. Therefore, it is possible for someone with access to the network to use network sniffing tools to gather information from that traffic. Some things that are available over the wire are file contents, user conversations, and even authentication credentials (usernames and passwords). If wire-level security is a concern, administrators should configure the server to use secure http (HTTPS) 55.

HTTP Cookies
PeerReview Complete uses two different cookies to track user activity: a session cookie; and a "remember me" cookie. The session cookie consists of a unique identifier assigned to the user at login time. Once logged in, this cookie allows the user to continue accessing PeerReview Complete without needing authenticate repeatedly. This cookie expires after 30 minutes of inactivity.

The "remember me" cookie is a longer lived cookie that, if present, reauthenticates the user if their session cookie expires. This cookie contains encrypted authentication information and if compromised, could allow an attacker to impersonate a user or potentially extract the user's password from the cookie. In high security environments, this cookie can be disabled by editing the context configuration file (`.../tomcat/conf/Catalina/localhost/ROOT.xml`) and removing the line that begins with "`<Valve className="com.smartbear.ccollab.auth.CollabFormAuthenticator"`".

© 2003-2011 SmartBear Software

The key used to encrypt the cookies is produced by a random number generator that is seeded with the random bytes in the "seed" attribute of the valve declaration. By default, the installer generates a pseudo-random seed at install-time, but administrators should feel free to update the seed at any time. Changing the seed will cause any existing cookies to fail to authenticate, but will not protect those cookies from being cracked by an attacker.

XMLRPC Security
Prior to version 5.0, PeerReview Complete relied on trusted clients to participate in authentication of users. As of 5.0, the default install is to not trust the clients and instead force them to provide credentials with each XMLRPC request. While this is more secure, and is the recommended configuration, we recognize that this configuration breaks older clients. So, we have implemented the following compromise:

1. Client compatibility is controlled by an application parameter configured in the context configuration file (`../tomcat/conf/Catalina/localhost/ROOT.xml`). The parameter name is "`client-compatibility`" and if the value is set to "`true`", older clients are allowed to connect to this server.
2. By default, upgrades from 4.0 will have client compatibility enabled -- that is, older clients will still be allowed to connect to the server. This is in recognition of the fact that server upgrades usually lead client upgrades by some period of time. We recommend that client compatibility be disabled as soon as clients can be upgraded.
3. By default, new installs will have client compatibility disabled. The assumption here is that new server installs will usually correspond to new client installs. Unless there are existing older clients (4.0 or earlier), it is not advisable to change this setting.

Note: This setting is explicitly a stop-gap for backwards compatibility. In the future, we expect to break backwards compatibility with 4.0 clients and always require preemptive authentication. However, this setting is expected to remain in place for all 5.x builds.

2.2.5 LDAP Authentication

By default the PeerReview Complete server authenticates users against the users in its database. For large organizations with hundreds or thousands of users in multiple product groups, it is simply impractical to add each would-be PeerReview Complete user to the database. For this reason, PeerReview Complete can integrate with an existing LDAP directory (or ActiveDirectory) to perform user authentication.

When LDAP authentication is configured, PeerReview Complete authenticates users attempting to login against their entry in the directory. When a user logs in 125 for the first time, a user account is created for them automatically in PeerReview Complete to store their user preferences. Users must be authenticated by logging into the PeerReview Complete server through the web GUI before the client is able to recognize the user and can be configured.

© 2003-2011 SmartBear Software

Warning: Only users with accounts can be assigned to roles within reviews, so it is not possible to add a user to a review who has not previously logged in to PeerReview Complete.

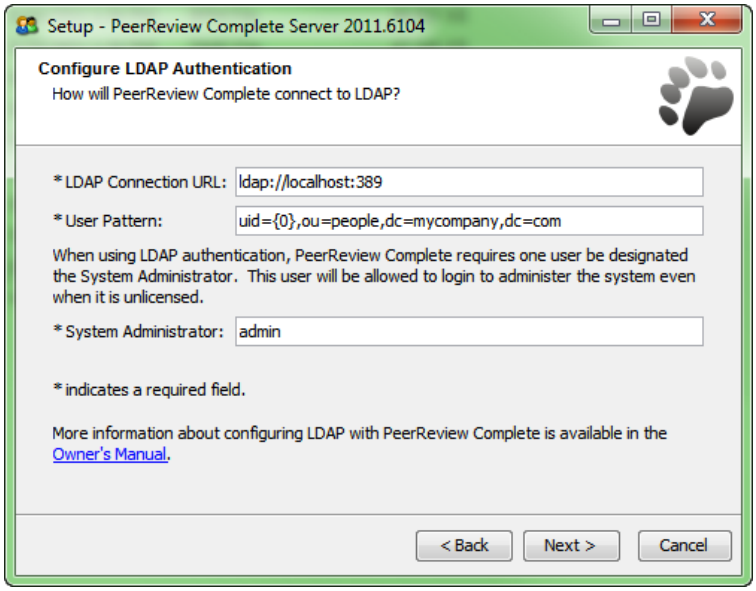
Internally, PeerReview Complete uses the Tomcat Servlet Container's JNDI Realm for LDAP authentication. For a detailed description of how it works and for complete configuration information, consult the JNDI Realm Documentation.

Note: LDAP Authentication is only supported in PeerReview Complete and is not supported by CodeReviewer. For a complete list of differences between the two products, please see the comparison page.

Configuring PeerReview Complete with LDAP

The installation wizard 31 provides a screen to perform basic configuration of LDAP authentication. This wizard minimally configures PeerReview Complete to use LDAP authentication. For advanced configuration, see Advanced Configuration 54 or contact Customer Support 69.

When LDAP authentication is selected, you are prompted for the following items:


Setup - PeerReview Complete Server 2011.6104
Configure LDAP Authentication
How will PeerReview Complete connect to LDAP?
* LDAP Connection URL:
ldap://localhost:389
* User Pattern:
uid={0},ou=people,dc=mycompany,dc=com
When using LDAP authentication, PeerReview Complete requires one user be designated the System Administrator. This user will be allowed to login to administer the system even when it is unlicensed.
* System Administrator:
admin
* indicates a required field.
More information about configuring LDAP with PeerReview Complete is available in the Owner's Manual.
< Back
Next >
Cancel

- LDAP Connection URL This is a URL where PeerReview Complete can connect to the LDAP server. The format of the url is: `ldap://servername:port` The standard default port for LDAP is `389`.
- User Pattern This string instructs PeerReview Complete how to locate a user within LDAP. The format of the string is the same as the LDAP distinguished name of a user with the username replaced with the string `{0}`.
- System Administrator The username of the system administrator. The system administrator will always be able to log in and administer license codes and user accounts. This account takes the place of the `admin` account when using internal authentication.

© 2003-2011 SmartBear Software

Active Directory

Microsoft Active Directory is an LDAP compliant directory and can be used to authenticate users to PeerReview Complete. However, in typical Active Directory environments, more configuration is needed than can be provided by the installation wizard. Specifically, the username that users log in with typically corresponds to the `sAMAccountName` attribute of their record. Because this is not part of their LDAP distinguished name (DN), PeerReview Complete cannot create a DN directly from the user-entered username to bind to the directory. Instead, PeerReview Complete must query the directory to get the user's distinguished name and then attempt the bind. To configure this, you will need to first install PeerReview Complete, specifying LDAP authentication and then manually edit the configuration file. The configuration file in question is *`installation-directory`*`/tomcat/conf/Catalina/localhost/ROOT.xml`. Locate the `Realm` element in the context file and replace it with the following:

```
<Realm className="org.apache.catalina.realm.JNDIRealm"
       connectionName="xxxx@xxx.com"
       connectionPassword="xxxx"
       connectionURL="ldap://xxxx.com:389"
       referrals="follow"
       userBase="CN=Users,DC=xxxx,DC=xxxx,DC=com"
       userSearch="(sAMAccountName={0})"
       userSubtree="true"
       allRolesMode="strictAuthOnly"
/>
```

You will need to update the attributes which have **`xxxx`**'ed out fields. The fields `connectionName` and `connectionPassword` define a user account the PeerReview Complete will use to connect to Active Directory to find the user records. If anonymous connections are allowed to your directory (not typical), then these attributes are not required.

> **Warning:** Modifying the ROOT`.xml` file will cause Tomcat to dynamically reload the PeerReview Complete application, terminating any active sessions. Changes to ROOT`.xml` should be done in the context of stopping and restarting the PeerReview Complete service (i.e., in a production environment coordinating the restart with user activity), regardless of whether the service itself is actually stopped and restarted, or just reloaded by Tomcat.

To use security groups or other user attributes to restrict access to PeerReview Complete, you can refine the `userSearch` query to return only users that meet your requirements.

The configuration examples above are a starting place for configuring Active Directory authentication. Additional configuration may be required because of specific configuration details of the directory. The JNDI Realm Documentation describes the different operational modes and explains each of the configuration attributes, and may be helpful in debugging the issue. The directory administrator that manages the directory will be a valuable resource in resolving the issue, either directly, or in conjunction with SmartBear's Customer Support team.

© 2003-2011 SmartBear Software

If the directory administrator is unavailable, SmartBear's Customer Support team can help you resolve configuration issues, but often do not have enough information about the directory schema, permissions, etc., to efficiently resolve issues. We will walk you through several basic configurations that we have seen work with other directories. Having an LDAP browser tool available (there are many good free and commercial browsers available) when you call will help answer some of the questions required to properly configure your server. In some cases, we will still need to discuss details with your directory administrator.

Secure LDAP (LDAPS)

Configuring PeerReview Complete to communicate securely with an LDAP server using LDAPS (LDAP over SSL) can be done, but requires manual configuration outside of the installer wizard. To configure LDAPS, first install PeerReview Complete configured for normal LDAP access. The service will start automatically upon completion of the install, so you will need to shut it down to continue configuration. Open the context configuration file (*`installation-directory`*`/tomcat/conf/Catalina/localhost/ROOT.xml`) in a text editor and find the Realm configuration element. It will look something like the following:

```
<Realm    className="org.apache.catalina.realm.JNDIRealm"
       connectionURL="ldap://localhost:389"
       userPattern="uid={0},ou=people,dc=mycompany,dc=com"
       allRolesMode="strictAuthOnly"   />
```

If you are configuring PeerReview Complete for use with Microsoft Active Directory using LDAPS, follow the Active Directory instructions [52] to make a best effort to configure the realm for your Active Directory server. Do not worry if you do not get it exactly right or cannot test the connection because the server refuses insecure connections. That issue can be resolved once connectivity is established.

To the realm configuration above (or your Active Directory realm configuration), you will need to add an attribute 'protocol' with the value 'ssl' and you will probably need to change the 'connectionURL' attribute to an LDAPS url. The updated configuration below highlights the changes:

```
<Realm    className="org.apache.catalina.realm.JNDIRealm"
       connectionURL="ldaps://localhost:636"
       userPattern="uid={0},ou=people,dc=mycompany,dc=com"
       allRolesMode="strictAuthOnly"
       protocol="ssl" />
```

Depending on the LDAP server's SSL certificates, this configuration *may* be enough to establish the connection. However, often companies generate their own SSL certificates signed by their own Certificate Authority (CA) certificate. Unless additional measures are taken, these certificates may not be trusted so PeerReview Complete will still not connect to the LDAP server. To establish trust, you will need to create a "key store" and import the public key of either the Certificate Authority or the public key of the LDAP server as a trusted certificate. Obtain from your LDAP or network administrator the appropriate certificate file. Then using Java's keytool (located in `$JAVA_HOME/bin`), import the certificate into a key store for PeerReview Complete using the following command:

```
keytool -import -alias ldapserver -keystore <collab install dir>/tomcat/conf/collab.ks -trustcacerts \
   -file <path to chain certificate file>
```

© 2003-2011 SmartBear Software

Important: You will likely be prompted to confirm the validity of the certificate. It is imperative for the security of the overall system that you verify the key matches the trusted material. Before accepting the certificate, you should contact the administrator that sent you the certificates and verify that the certificate fingerprints that you see match the certificate fingerprints that they intended to send you. For more information on why this step is important, see the note in the keytool documentation.

The final step is to configure PeerReview Complete to use the newly created keystore. Open `installation-directory/ccollab-server.vmoptions` in a text editor and add the following lines:

```
-Djavax.net.ssl.trustStore=<installation-directory>/tomcat/conf/
collab.ks
-Djavax.net.ssl.trustStorePassword=<the password>
```

Upon restart, the PeerReview Complete service should be connecting to the LDAP server via SSL. If you are still getting errors, check that the other LDAP configuration options have been configured correctly. If you are using Active Directory, it is now worth revisiting the Active Directory configuration above.

Finally, a note on troubleshooting SSL connections: adding the following line to the `ccollab-server.vmoptions` file will enable Java's network debug logging.

```
-Djavax.net.debug=all
```

Upon restarting PeerReview Complete, this information will be written to `<installation-directory>/output.log`. If you need assistance interpreting this log, contact SmartBear Customer Support.

Note: Do not run in a production environment with network debug logging enabled. This will severely impact the performance of the system and will also consume vast quantities of disk space.

Advanced Configuration

The basic configuration provided by the installer sets up the simplest possible LDAP configuration. Many more sophisticated configurations are possible, but require manually editing the context configuration file (`installation-directory/tomcat/conf/Catalina/localhost/ROOT.xml`).

Users From Different Organizational Units

Sometimes it is necessary for users from different organizational units to have access to PeerReview Complete. If you are using direct-bind configuration using the `userPattern` attribute, you can change the configuration to use `userPatternArray` instead where the value is a colon-separated list of user patterns. For example, if there are users in `ou=foo,dc=mycompany,dc=com` and `ou=bar,dc=mycompany,dc=com` identified by their `uid` attribute, you can set the `userPatternArray` attribute as follows:

```
userPatternArray="(uid={0},ou=foo,dc=mycompany,dc=com):(uid=
{0},ou=bar,dc=mycompany,dc=com)"
```

Likewise, if you are using search to locate users (as in Active Directory Configuration 52) you can use the LDAP filter operators | and & (SGML encoded as &) to form compound filters. Taking the same example:

© 2003-2011 SmartBear Software

```
userSearch=(|(uid={0},ou=foo,dc=mycompany,dc=com)(uid={0},
ou=bar,dc=mycompany,dc=com)
```

Restricting Access

If you need to restrict access to PeerReview Complete, we recommend that you configure your Realm definition to use `userSearch` for searching for user accounts rather than `userPattern` for direct bind (see Active Directory Configuration 52 for examples). With `userSearch`, you can expand your search criteria to include only members of the specified group or groups. For example, the following `userSearch` would restrict access to only members of the `ccusers` security group:

```
userSearch="(&amp;(memberOf=CN=ccusers,OU=Security Groups,
OU=Accounts and Groups,DC=xxxx,DC=xxxx,DC=com)
(sAMAccountName={0}))"
```

Please note that the ampersand (&), needs to be SGML encoded to `"&"` because the configuration file is an XML document.

Previous versions of PeerReview Complete advised administrators to manually edit the deployment descriptor (`web.xml` file) to include an role-name in the `auth-constraint`. While this method will work to restrict access to PeerReview Complete, it is not preserved after a server upgrade, so administrators were required to edit the deployment descriptor at each upgrade. The `userSearch` method described above is preserved on upgrade, so upgrades are much simpler.

Administrator Rights

When users first log in to PeerReview Complete, their user account is created automatically, as a standard user account. It is possible to have users in certain roles (see Restricting Access above) to automatically receive administrator privileges. This configuration is done in the *`installation-directory`*`/tomcat/conf/Catalina/localhost/ROOT.xml` file.

Near the bottom of the ROOT`.xml` file, there is a section of server parameters (`Parameter` XML elements). These are name-value pairs of configuration options. To configure certain LDAP groups to be assigned administrator rights, create an `admin-roles` parameter and for its value specify a comma-separated list of LDAP group names. New users that are members of the specified LDAP groups will be given administrator privileges when their accounts are created. Users that already have accounts in PeerReview Complete will not automatically be promoted to administrator. Those privileges will need to be assigned manually from the `Admin` screen.

Warning: After making changes to the ROOT`.xml` file, you will need to restart the PeerReview Complete service for the changes to take effect.

2.2.6 Configuring HTTPS

The basic PeerReview Complete installation configures the server to handle requests over standard HTTP. In many environments this is sufficient as the network is trusted. However some organizations require that all network applications are secured with Transport Layer Security (TLS) or Secure Sockets Layer (SSL). PeerReview Complete supports HTTP over TLS (or HTTPS), but this requires additional

© 2003-2011 SmartBear Software

manual server configuration:

What you need

Certificates

In order to authenticate to clients, the PeerReview Complete server must have a "Certificate" that serves as proof of identity. Certificates come in two forms: Certificate Authority (CA) signed certificates and "self-signed" certificates. CA signed certificates provide an additional level of security because they can be automatically verified and do not rely on human verification. By providing you a certificate, the Certificate Authority is vouching for your identity. Software systems such as web browsers and the Java Runtime Environment (JRE) include the public keys of the trusted Certificate Authorities that are used to verify server certificates were vouched for by a trusted CA. To acquire a CA signed certificate, contact the appropriate person in your IT department.

Self-signed certificates have the advantages of being free and easy to generate. The disadvantage of self-signed certificates is that they cannot be automatically verified. Instead, their security relies on users verifying the certificate signature against a signature obtained through another, ideally secure, channel.

Keystores

A keystore is Java's mechanism for storing cryptographic keys and certificates. The persistent form of a keystore is a password protected file on disk. A keystore may contain "key entries" that allow applications with access to the keystore to authenticate themselves to users or other services. A keystore is also used to store "trusted certificate entries" that contain the public keys of services that are trusted and certificate authorities that are trusted to vouch for services.

To enable HTTPS in PeerReview Complete, you will need to create a keystore with the server's certificate (either CA- or self-signed) and if CA-signed, the certificate authority's "chain certificate". Follow the instructions below to create a keystore with the type of certificate you will be using to secure PeerReview Complete.

Getting Keystores and Certificates

Importing CA-signed Certificates

If you are using a CA-signed certificate, you will need the server certificate and the CA's chain certificate (or root certificate). The CA chain certificates are publicly available, but the instructions for acquiring them vary by certificate authority. To get the CA chain certificate, inquire of the IT person in your organization responsible for procuring SSL certificates for services. Once you have the certificates, import them into a new keystore using the Java "`keytool`" program (located in `$JAVA_HOME/bin`).

To import the CA chain certificate:

```
keytool -import -alias root -keystore <collab install dir>/tomcat/conf/collab.ks -trustcacerts
   -file <path to chain certificate file>
```

To import the server certificate:

```
keytool -import -alias tomcat -keystore <collab install dir>/tomcat/conf/collab.ks -trustcacert
   -file <path to server certificate file>
```

Remember the keystore password you select, as you will need this in the next step.

© 2003-2011 SmartBear Software

Creating a Keystore With a Self-signed Certificate

If you are using a self-signed certificate, you can create the certificate and populate a keystore with a single invocation of Java's `keytool` (located in `$JAVA_HOME/bin`). Simply run the following command:

```
keytool -genkey -alias tomcat -keyalg RSA -keystore <collab install dir>/tomcat/conf/collab.ks
```

`Keytool` will ask you a variety of questions about the certificate you are generating. Be sure to remember the keystore password, as you will need it when configuring the server.

After you have generated the keys for the server, you will need to distribute its signature to users so they can validate the certificate when asked. To do this, first export the certificate to a file:

```
keytool -export -keystore <collab install dir>/tomcat/conf/collab.ks -alias tomcat > collab.cer
```

Then print the certificate information that you will need to distribute. Of particular interest are the fingerprints. To do this:

```
keytool -printcert -file collab.cert
```

This will print something like the following:

```
Owner: CN=PeerReview Complete, OU=SmartBear, O=SmartBear, L=Austin, ST=TX, C=US
Issuer: CN=PeerReview Complete, OU=SmartBear, O=SmartBear, L=Austin, ST=TX, C=US
Serial number: 463251eb
Valid from: Fri Apr 27 14:41:31 CDT 2007 until: Thu Jul 26 14:41:31 CDT 2007
Certificate fingerprints:
           MD5:  67:D7:74:5E:72:9D:B2:82:88:3F:33:AA:A0:41:01:F0
            SHA1: E2:4A:1F:9B:9A:38:0F:6B:7B:33:12:73:1B:50:76:30:AC:A6:B2:EA
```

Existing Keys and Certificates

In some cases, you may already have a signed certificate (generated by openssl, for instance) and so don't want to use keytool to generate a key and certificate for you. In those cases, you still must put the key and certificates into a keystore so that the PeerReview Complete server can use them.

You import it into the keystore by first putting it into a pkcs12-format keystore:

```
openssl pkcs12 -export -in certified_collab_certificate.pem \
       -inkey  private_collab_key.pem \
       -out mystore.p12 -name tomcat \
        -CAfile CA_chain_certificate.cer \
       -caname root -chain
```

and then import the pkcs12 keystore into the java keystore by doing:

```
keytool -importkeystore \
  -destkeystore tomcat/conf/collab.ks \
  -srckeystore mystore.p12 -srcstoretype PKCS12
```

Note that of course the .pem, .cer, and .p12 files may all be referred to using full path name.

Configuring the PeerReview Complete Server

The final step in configuring the PeerReview Complete server is to instruct the server to use SSL and tell it what certificates to use. This is done by editing the `server.xml` file located in `<install directory>/tomcat/conf`. In server.xml, locate the "Connector" element and replace it with the following:

© 2003-2011 SmartBear Software

```
<Connector  protocol="org.apache.coyote.http11.Http11Protocol"
        port="8443" minSpareThreads="5" maxSpareThreads="75"
        enableLookups="true" disableUploadTimeout="true"
       acceptCount="100"  maxThreads="100"
        scheme="https" secure="true" SSLEnabled="true"
        keystoreFile="conf/collab.ks" keystorePass="<the keystore password>"
       clientAuth="false" sslProtocol="TLS"/>
```

Note, you will need to replace the keystore password in the above XML snippet. That is all the required configuration. Restart the server and it will be operating over SSL on port 8443.

Be sure to update the External URL 73 in the server settings to reflect the correct https URL.

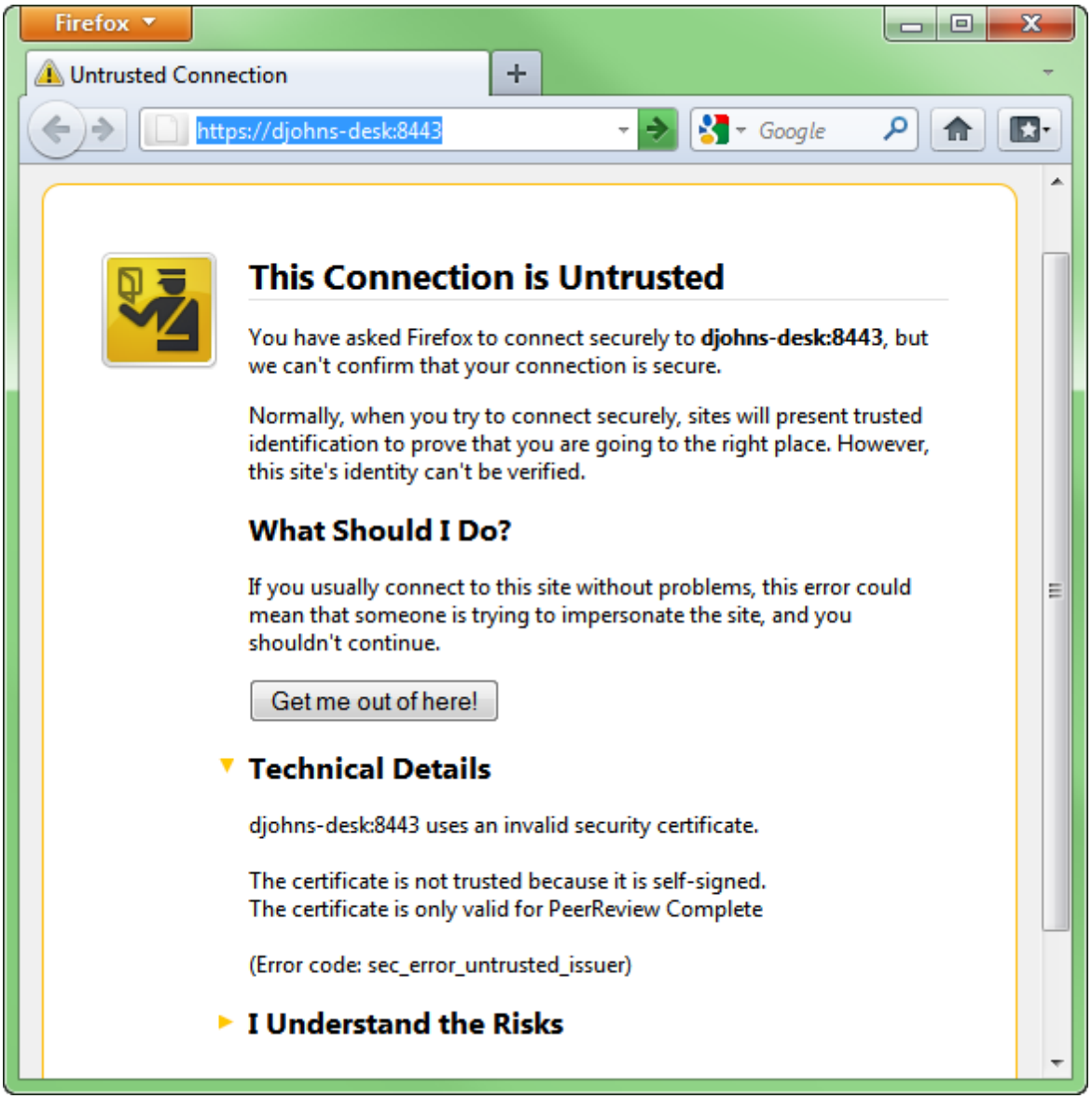
Impact on Clients

If the certificate used to secure PeerReview Complete is a CA signed certificate from a trusted CA, there will be very little impact on users. The only thing that they must do is configure their clients 187 with the correct HTTPS url.

If you are using a self-signed certificate or the certificate cannot be automatically verified, users will be asked to verify the certificate by comparing its signature to the published signatures for the service. The exact nature of this confirmation varies by client, but the process is the same: look at the certificate the server presents and confirm that its signature matches the published signature.

In the web browser, the certificate confirmation dialog looks something like this:

© 2003-2011 SmartBear Software


Firefox
Untrusted Connection
https://djohns-desk:8443
Google
This Connection is Untrusted
You have asked Firefox to connect securely to djohns-desk:8443, but we can't confirm that your connection is secure.
Normally, when you try to connect securely, sites will present trusted identification to prove that you are going to the right place. However, this site's identity can't be verified.
What Should I Do?
If you usually connect to this site without problems, this error could mean that someone is trying to impersonate the site, and you shouldn't continue.
Get me out of here!
Technical Details
djohns-desk:8443 uses an invalid security certificate.
The certificate is not trusted because it is self-signed.
The certificate is only valid for PeerReview Complete
(Error code: sec_error_untrusted_issuer)
I Understand the Risks

The above screenshot is from Firefox, but other browser users will see a similar dialog. The users will need verify that the fingerprint (or signature) of the certificate by comparing it to the signature of that you distributed for the server.

When the command line client fails to validate the server's certificate, it also offers the user a chance to manually validate the certificate. This conversation looks like the following:

```
~bdurette/work> ccollab login http://myserver.com bdurette password

The SSL certificates sent by the server could not be validated:
No trusted certificate found

It is possible, though unlikely, that someone may be trying to
intercept your communication with this service.  If you have
questions about the validity of the certificate, please check with
```

© 2003-2011 SmartBear Software

```
your  administrator.

What would you like to do?
P: Print the certificate chain.
A: Accept the certificate permanently (automatically accept this certificate in the future).
T: Temporarily accept this certificate (confirm again upon next connection).
R: Reject the certificate.
P
Subject: CN=PeerReview Complete, OU=SmartBear, O=SmartBear, L=Austin, ST=TX, C=US
Issuer: CN=PeerReview Complete, OU=SmartBear, O=SmartBear, L=Austin, ST=TX, C=US
Serial Number: 463251eb
SHA1   Fingerprint:   e2:4a:1f:9b:9a:38:0f:6b:7b:33:12:73:1b:50:76:30:ac:a6:b2:ea
MD5   Fingerprint:   67:d7:74:5e:72:9d:b2:82:88:3f:33:aa:a0:41:01:f0

What would you like to do?
P: Print the certificate chain.
A: Accept the certificate permanently (automatically accept this certificate in the future).
T: Temporarily accept this certificate (confirm again upon next connection).
R: Reject the certificate.
A

PeerReview Complete connection information set.   Connection is good.
```

Certificates that are accepted permanently are stored in a local keystore in `$HOME/.smartbear/trustedcertificates`. This file can safely be deleted to remove all trusted certificates. If this is done, the user will have to reauthorize any certificates that are self-signed (or signed by a CA that's not in the Java trusted keystore).

Additional Resources

- Tomcat SSL Configuration HOW-TO
- Tomcat HTTP Connector Configuration
- `keytool` - Java Key and Certificate Management Tool

2.2.7 JMX Monitoring

You can monitor many aspects of the PeerReview Complete server via JMX, Java's standard, built-in network monitoring system.

Setting up a basic JMX system

To enable JMX you must add Java properties to the server startup script. To do that, edit the file `ccollab-server.vmoptions` which is found inside the server installation directory.

Inside this file, add the following lines:

```
-Dcom.sun.management.jmxremote=true
-Dcom.sun.management.jmxremote.port=54321
-Dcom.sun.management.jmxremote.authenticate=false
-Dcom.sun.management.jmxremote.ssl=false
```

© 2003-2011 SmartBear Software

The first line enables JMX. The second line sets the TCP/IP port number for the JMX protocol, and of course you can use a value other than the example `54321`. The last two lines disable JMX authentication and security.

> **Note:** You must restart the PeerReview Complete server before these settings take effect.

You can now use any JMX console software to access the PeerReview Complete JMX variables. Sun's Java installation ships with a simple JMX console which can be found in `[Java-Install-Dir]/bin/jconsole.exe`. If you use Sun's console, select the "Remote" tab in the connection dialog and specify the host name, the port given in the configuration above. Leave "User Name" and "Password" blank.

Supported JMX data
There are various things you can monitor with JMX, and this manual cannot cover them all.

The data specific to PeerReview Complete can be found under "MBeans," then under the root tree node `"com.smartbear.ccollab."` Under there you will find one child tree node for each installation of PeerReview Complete, named by the web page it's installed under. The default is just a simple forward slash. Under that node is various data.

Of particular note is `"ApplicationStateCache"` and `"ServletState."` This contains basic information about logins, licensing, versioning, and product usage.

More complex JMX server configurations
A full description of how to configure JMX is outside the scope of this manual. It is completely standardized and documented on Sun's web site. A good resource to get you started is this:

http://java.sun.com/j2se/1.5.0/docs/guide/management/agent.html

2.2.8 Technical Specifications

This section describes the technical aspects of the PeerReview Complete server. This information is not generally needed for server administration.

Server Technology
The web server is Apache Tomcat v6.0.20. PeerReview Complete itself is a collection of standard J2EE servlets, packaged in the standard WAR format that can be used inside any J2EE servlet container. Tomcat is the only servlet container specifically supported by SmartBear.

Tomcat is a sophisticated, scalable stand-alone web server with support for HTTPS 55, LDAP authentication 50, advanced HTTP protocol options, database pools, load-balancing, and more.

For more information about how PeerReview Complete is configured under Tomcat, see the Tomcat Configuration Reference.

© 2003-2011 SmartBear Software

Starting and Stopping the Server

Windows

The server is a Windows Service. This means you can start, stop and restart the server just like any other service. The typical way is through the "Services" Control Panel. You can also use "`net start ccollab-server`" and "`net stop ccollab-server`" from the command-line.

Unix / Mac OS X

The server is a shell script in the installation directory called `ccollab-server`. You can use the usual "`start`" and "`stop`" directives, which also means the script can be used directly in an `init.d` system.

Recommended Hardware

For trials, you can use any hardware available. We have lots of customers who have done pilots on "servers" such as laptops and regular workstations. For small groups (30 users or less) doing small reviews, the server is not demanding on CPU, memory, disk access, or database access; if you run the server on a workstation, you'll never know it's there.

For permanent installations, and to support hundreds or thousands of users, we recommend the following minimum hardware:

- Dual XEON 3.5ghz processors
- 2G RAM
- 10,000 RPM hard drives, preferably SCSI.
- 100GB storage space (this number varies a lot based on usage patterns)
- Some kind of drive backup hardware (RAID 0 is fine)
- Windows XP Server, Linux, or Solaris
  (If Windows, *must* be a server edition, not workstation edition)

We run our own scalability tests using this hardware. Our standard "smoke test" is 500 caffeinated users, most hitting Review Summary 148 and Side-by-Side 157 pages, a few in administrative 73 and reporting 176 pages. This represents far more activity than 500 real users.

We currently have zero open scalability issues for customers with over 1000 users with hardware at least as capacious as the set-up given above.

Performance Tuning

General tips for making the server run as fast as possible:

- Use the latest version of the server and client.
  We're constantly improving speed bottlenecks as they are reported. Upgrade to the latest version, or scan the version history 462 to see if any speed improvements were made.
- Increase the "chat refresh interval 75" setting. This will make the chat pane update more slowly but can reduce stress on the server by many times. The default value is 5 seconds, but 15, 30, or even 60 seconds can work and reduce server load.

© 2003-2011 SmartBear Software

- Make sure nothing else is running on the PeerReview Complete server.
  Often another server or process running on the same machine as our server will take up CPU cycles or slam the hard drive. For small installations it's quite common (and appropriate) for PeerReview Complete to share a server, but for larger installations or when you're trying to squeeze out more performance you should dedicate a server. And this means really dedicate, not just its own `VMWare` instance.
- Increase the number of allowed database connections (instructions here 71).
  Even if you're not getting errors on the web page about running out of connections, the web server threads might still be waiting for database connections to become available. (Error messages don't appear until those waits start timing out.)
- Increase the speed of the connection between the database and the server.
  If your database isn't on the same machine, can you get gigabit ethernet? Can you put the database on the same machine as our server?
- Increase the number of server threads (instructions here 71).
  Just as with database connections, more server threads means less waiting for browser connections on a busy server.
- Don't run inside a virtual environment.
  Environments such as `VMWare` can impose as much as a 40% speed hit 63 (as shown in our own in-house tests). If you want maximum performance don't use extra layers between our server and the hardware.
- Increase the max number of simultaneous connections in your browser.
  Both IE and Firefox default to just two simultaneous connections to a server. You can increase this to 4 or more. This won't help if you have a very slow connection (e.g. a modem), because slow connections are probably already maxed out.
- Make sure caching is enabled in your browser.
  Sometimes a browser's static cache is disabled; make sure it's enabled (at least for connections to our server). We have lots of styles, Javascript, and images that normally are cached.

Running under `VMWare`

In our tests, the server performs at 40% capacity when running under `VMWare.` Our tests were performed against `VMWare Workstation v5.5` which itself was running under Windows XP.

Whenever you see performance or scalability numbers in this manual, you should mentally scale capacities by 40% when considering a `VMWare` or other virtualization environment. Your mileage may vary, and we're not experts in virtualization.

Monitoring

You can monitor our application like any other. Like many enterprise-ready Java applications, you can monitor using JMX 60.

2.2.9 High Availability Best Practices

The PeerReview Complete server is generally very stable and resilient. In most environments it's appropriate to simply try configuration changes or upgrades in-place (backup 46 first, of course).

However if the server is in constant use by a large number of users, or if PeerReview Complete is vital for your business processes, it may be a good idea to be more cautious. To ensure that a change won't cause a problem on your production PeerReview Complete server, it's best to try it first on a test server.

© 2003-2011 SmartBear Software

Creating a Test Server

A test server should mirror your production server as closely as possible so that the results of the change are reproducible. Follow these steps to create a test server:

1) Provision a machine - this should be physically different hardware than the machine that is running your production PeerReview Complete server, and ideally it should have the same specifications. It does not have to be a perfect match, but the closer you get it to the production server the more confident you will be when testing the change.

   a) Hardware - this is not that important to get exactly identical unless you are concerned about performance issues.

   b) Operating System - this is important. Make sure you install the same operating system version, especially if you're using Windows.

   c) Java Virtual Machine - this is vitally important. Make sure that you install exactly the same version of Java.

   d) Network Configuration - this is usually not that important unless your change involves communicating with an external system like LDAP 50, a database service running on a different machine, or configuring https 55.

   e) Database - this is very important, especially when testing upgrades to new versions of PeerReview Complete.

> **Warning:** The test server *must* have it's own database. Do not run the test server with your production database!

2) Install the PeerReview Complete server

   a) Download the installer for *exactly* the same version as your production server.

   b) Run the installer. You can click "next" through all the screens and accept the default settings, because those will be overwritten anyway in step 4 65.

   c) When you click "Finish" in the installer it will open a browser to a page that says "Initialize database". DO NOT press the button, instead close the browser.

   d) Shut down the PeerReview Complete server.

3) Copy your content-cache - this directory stores the content of the files being reviewed and can become very large. The directory is configured in the `installation-directory/tomcat/conf/Catalina/localhost/ROOT.xml` file and defaults to `installation-directory/tomcat/collaborator-content-cache`. Copy the entire directory to the test server in the same location.

   a) You can skip this step and your content-cache will simply be empty. Everything will work except when you look at the content of files in old reviews you will see a message "*This content has been archived by the Administrator*".

   b) Another option is to use a network share to your production server's content-cache, mounted to the test server's file system. While not ideal this is fairly safe because PeerReview Complete only ever writes to the content-cache, never deletes (unless you explicitly use the archive 122 feature).

© 2003-2011 SmartBear Software

4) Copy your configuration

   a) Copy the following files from your production server to your test server:

      i) *installation-directory*/ccollab-server.vmoptions

      ii) *installation-directory*/tomcat/conf/collab.ks (may not exist)

      iii) *installation-directory*/tomcat/conf/Catalina/localhost/ROOT.xml

   b) Edit the "*installation-directory*/tomcat/conf/Catalina/localhost/ROOT.xml" file using a text editor and change the database information (url, username, password) to point to the test database.

4) Copy your database data - migrate a copy of the data from your production database to your test database as described in Backup/Migration 46.

5) Start up PeerReview Complete.

   a) Immediately disable email notifications 99 so that your users don't get double notifications from your production server and your test server.

   b) Click around and make sure everything works. Note that licensing isn't an issue - since you've migrated your database your license will work on the test server.

Testing a Change

Once you've set up your test server you can use it to test any change you are considering to PeerReview Complete, especially upgrades to new versions or global configuration changes.

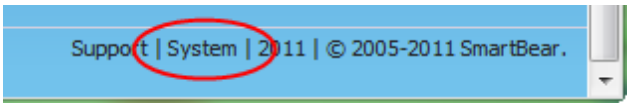
One good way to test the server is the built-in diagnostic tool. Run the diagnostics 65 before and after executing the change and look for any difference in the results.

2.2.10 Diagnostics

The PeerReview Complete server offers several diagnostics you can run to check the health of your installation. A failure result from one of the diagnostics does not necessarily represent an actual error, but may assist you and/or SmartBear technical support in debugging a problem with your server.

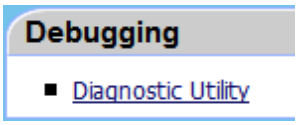
Accessing Diagnostics

To get to the diagnostics go to the PeerReview Complete web server and click the `System` link at the bottom of the screen:



Scroll down to the "Debugging" section and click on "Diagnostic Utility".

© 2003-2011 SmartBear Software



Running Diagnostics
To run a diagnostic click on one of the links, or click on the *RUN ALL DIAGNOSTICS* link to run all diagnostics. Some of the diagnostics take a long time to complete, and may impact the performance of your server while the diagnostic is running. See notes for each diagnostic below.

Verify Database Schema
This diagnostic checks the schema of your database to make sure that it is what PeerReview Complete expects. This diagnostic runs very quickly and will not impact performance on your server.

Fix Database Data
This diagnostic looks for invalid data in the database and fixes the data to be inside allowable ranges. If you have a large database this diagnostic may take some time to complete and could impact performance on your server while it is running. Generally it is only necessary to run this diagnostic if you have been told to do so by Smart Bear technical support.

Fix Duplicate Users Differing By Case
This diagnostic looks for users whose logins differ only by case and merges them to a single user account. This is used to repair problems caused by a bug in older versions of the PeerReview Complete server that allowed users to be created whose logins differed only by case. This diagnostic usually runs quickly and will not impact performance on your server.

Database Integrity
This diagnostic checks that database data is self-consistent and sensible. Some failures in this diagnostic are common and do not necessarily represent real errors. This diagnostic may take a long time to run (scales linearly with size of your database) and will impact performance on your server while it's running so it's best to *run it only when the server is not in use*.

Content Cache Integrity Check
This diagnostic checks that the on-disk file content cache is present, readable, and that all content therein is properly indexed. This diagnostic takes a long time to run (scales linearly with number and size of files in your content cache).

Content Cache Cleanup
This diagnostic deletes files from the on-disk file content cache that is not properly indexed, meaning the MD5 of the file content does not match the name of the file. This diagnostic takes approximately the same time to run as the Content Cache Integrity Check. You should back up your Content Cache before running this diagnostic.

© 2003-2011 SmartBear Software

2.2.11 Logging

PeerReview Complete servers log messages to several log files which are user configurable (with two exceptions). The configurable log files are all, by default, located in `<installation-directory>/tomcat/logs`. They are `collab.log` (and numbered generations of same when logs roll) for application level logging; tomcat.log for the application server logs; and `license.log` which logs license usage events, such as user login and logout (if enabled). The logs that do not allow user configuration are located in the installation directory and are named output.log and error.log which capture the standard out and standard error streams from the application. Normally, PeerReview Complete does not write to standard out or error, but some of the libraries that it uses do. Occasionally, these logs will be helpful for debugging issues in one of those libraries.

Configuring Logging Using Configuration Files

PeerReview Complete uses a sophisticated logging system, log4j, to manage the logging. Because of the breadth of configuration options available in log4j, it is not possible to document everything here (the log4j documentation is available at http://logging.apache.org/log4j/1.2/manual.html). However, in this section, we will cover some common scenarios.

The configuration file for `collab.log` and `license.log` can be found at `<installation-directory>/tomcat/webapps/ROOT/WEB-INF/classes/log4j.properties`. The file is a plain text Java properties file and can be edited with any text editor. Changes will take effect when the server is restarted.

> **Note:** Edits to the log4j.properties file will not be saved between updates to PeerReview Complete. Upon editing this file, it is a good idea to save a backup of the file in a location outside the PeerReview Complete installation tree.

Basic Debug Logging

The most common configuration change that system administrators will make is to turn on debug logging. This can be useful when working with SmartBear technical support to resolve issues (though this is not always required). To enable debug logging, find the first non-comment line in the file:

```
log4j.rootLogger=info, PeerReview Complete
```

This line configures the base level logging to "info" level and attaches an appender named "PeerReview Complete" that will be configured later in the file. To enable debug logging, simply replace "info" with "debug" and restart the server. Do not change the name of the appender as this will result in the log configuration not being available via JMX. Running a production system at debug logging level can cause performance issues, so it is not recommended unless specifically directed by SmartBear technical support.

Log Rolling

By default the `collab.log` file is rolled when it reaches 10Mb and saves 10 generations of files, for a total of 110Mb of disk space potentially consumed. Both of these values can be configured in the log configuration file. The following lines specify these values:

```
log4j.appender.PeerReview Complete.MaxFileSize=10MB
log4j.appender.PeerReview Complete.MaxBackupIndex=10
```

© 2003-2011 SmartBear Software

System administrators may change these values as necessary to capture additional logging information or conserve disk space as necessary.

Enabling License Logging

PeerReview Complete can log license events (users logging in and out) to help you understand your license consumption. By default this is disabled as it is normally not necessary. The system status 123 page includes graphs which are helpful for understanding the total license consumption. However, drilling down to find what users are using the system when requires turning on license logging. To do this, change the license logging level from "warn" to "info":

```
log4j.logger.com.smartbear.ccollab.license=warn, PeerReview
CompleteLicense
```

The license log appender is also configured to roll like the `collab.log` and this configuration can be changed in a similar manner. However, it is important that you not change the appender name from "PeerReview CompleteLicense" as this will result in the log configuration not being available via JMX 68.

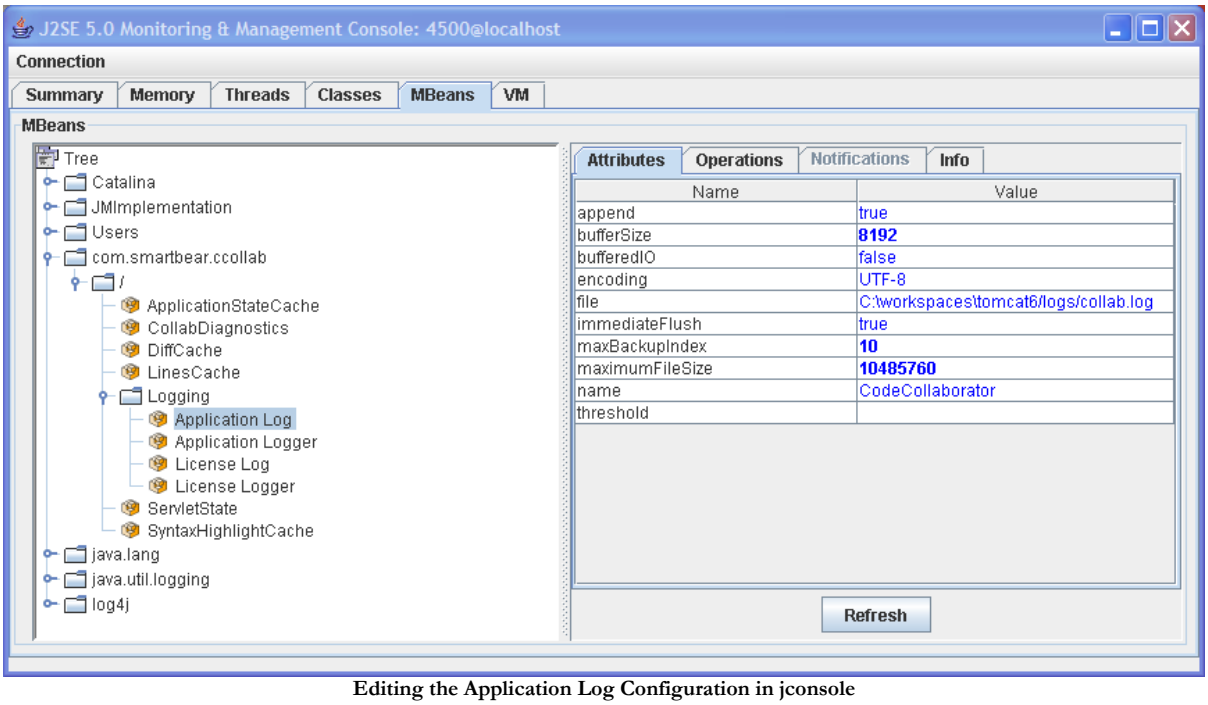
Controlling Logging with JMX

If the PeerReview Complete server has JMX management enabled 60, the logging level can be changed at runtime. All of the log configuration MBeans are available at com.smartbear.ccollab:/:Logging. Changing the "priority" on "Application Logger" or "License Logger" will change the base level of logging for the application and the license log respectively.

The appenders are found under the names "Application Log" and "License Log". The max size, number of backups, and even the file names, can be changed in JMX. The changes will not take effect until the "activateOptions" operation is invoked. In jconsole operations are found in a separate tab.

© 2003-2011 SmartBear Software

Note: Changes to logging made via JMX are not persisted across server restarts.



Editing the Application Log Configuration in jconsole

2.2.12 Troubleshooting

When something goes wrong, refer to this section.

Known Issues

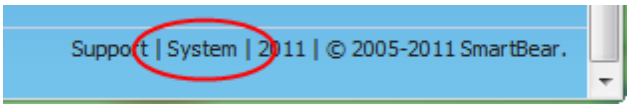
Check the Known Issues Appendix 460 to see whether SmartBear already knows about this issue.

Version History

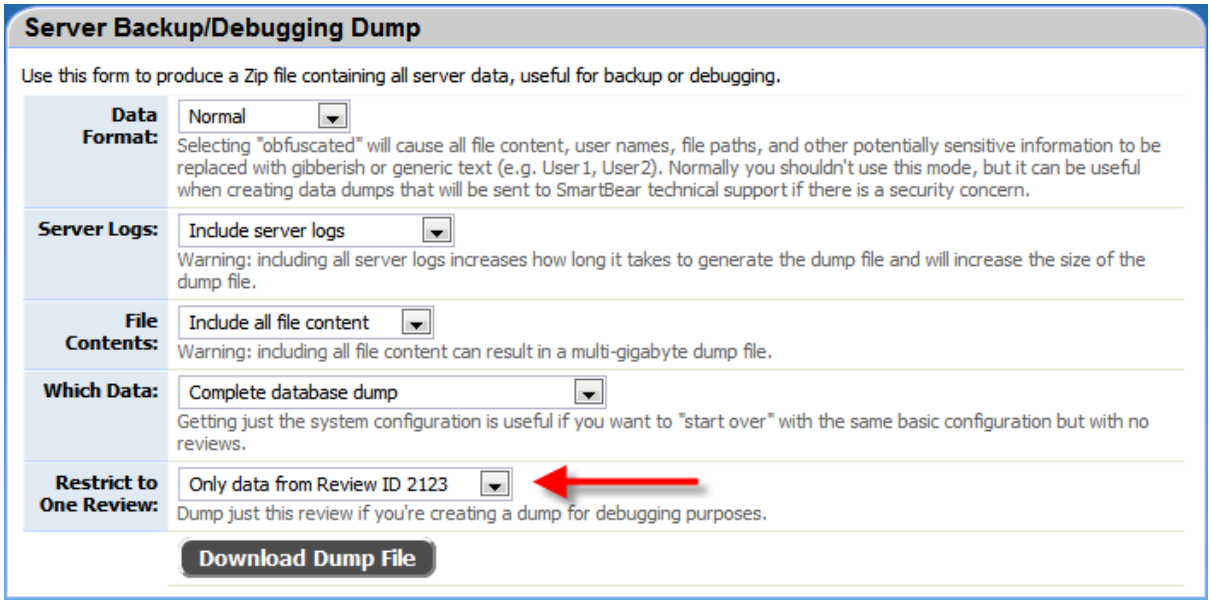
Check the Version History on the SmartBear website to see if this issue has been resolved in a later release.

Getting server information

To get debugging information about the server, go to the PeerReview Complete web server and click the `System` link at the bottom of the screen:



Downloading server dumps for SmartBear **technical support**
Sometimes it will be useful for SmartBear technical support to get a "dump" of your server configuration and database data. You do this by filling out the form at the top of the `System` page.


Server Backup/Debugging Dump
Use this form to produce a Zip file containing all server data, useful for backup or debugging.
Data Format:
Normal
Selecting "obfuscated" will cause all file content, user names, file paths, and other potentially sensitive information to be replaced with gibberish or generic text (e.g. User1, User2). Normally you shouldn't use this mode, but it can be useful when creating data dumps that will be sent to SmartBear technical support if there is a security concern.
Server Logs:
Include server logs
Warning: including all server logs increases how long it takes to generate the dump file and will increase the size of the dump file.
File Contents:
Include all file content
Warning: including all file content can result in a multi-gigabyte dump file.
Which Data:
Complete database dump
Getting just the system configuration is useful if you want to "start over" with the same basic configuration but with no reviews.
Restrict to One Review:
Only data from Review ID 2123
Dump just this review if you're creating a dump for debugging purposes.
Download Dump File

Data Format
You can dump the entire database in a normal format, or set the "Data Format" field to "Obfuscated," which will dump the data after passing through a filter that converts potentially sensitive information into unhackable text. For example, users are renamed "user1," "user2," and changelist text is replaced by the MD5 of that text. If file contents are included in the dump, each line of file content is replaced with the MD5 of that line to enable us to reproduce exactly the right line numbers and diffs you're seeing. This can be used to send us data when you're worried there might be sensitive information.

If you're worried about whether we're really cleaning all the data, you can check out the dump file yourself. It's just a ZIP file containing XML with information, so it's easy to inspect. In fact, if you still see some data you'd like to change or delete, you can just do that in the XML, re-zip the file, and send it to us.

Server Logs
You can choose to include all server logs or none at all.

File Contents
You can choose to include all file content or none at all.

Which Data
Here you can choose whether you'd like to perform a complete database dump or just include the system configuration.

© 2003-2011 SmartBear Software

Review Dumps
If just one review is broken, you can send us a dump of just the one review. To do that, first go to the review in question, then click the `System` link as before. Now you will see an additional field in the form for downloading review data (which includes all the file content as well).

Diagnostics

PeerReview Complete has some built-in diagnostics [65] that may be helpful in diagnosing a problem. You may be prompted to run one of these diagnostics by SmartBear technical support.

The server is running slowly
Check out our performance tuning [62] suggestions and use recommended hardware [62].

Email is not working
If email isn't working, here are some things to try:

- Make sure the SMTP Host and SMTP Port settings [99] are correct.
- Try sending a test email [99]. Look in the server logs [72] for error messages.
- If you just performed a database migration/restore [46], email is automatically disabled. You need to enable it in the settings [99].

Getting "Idle Timeout Waiting for Object" Error
This error is shown when the database connection pool runs out of connections to the database.

The fix is to increase the number of connections allowed in the pool. The server is capable of running with many more connections than the default; we keep the default fairly low so that in smaller installations we're not taking up too many database resources.

Fix this by going into the Tomcat servlet context XML file located here:

`installation-directory/tomcat/conf/Catalina/localhost/ROOT.xml`

Edit the property called `maxActive` and increase the number of connections. Even doubling this number is normal. A rule of thumb is to have 3 times the database connections (and server threads) as the number of simultaneous users you have under the biggest load.

The PeerReview Complete server must restart for this change to take effect. You do not need to do anything to your database server.

> **Warning:** Modifying the ROOT`.xml` file will cause Tomcat to dynamically reload the PeerReview Complete application, terminating any active sessions. Changes to ROOT`.xml` should be done in the context of stopping and restarting the PeerReview Complete service (i.e., in a production environment coordinating the restart with user activity), regardless of whether the service itself is actually stopped and restarted, or just reloaded by Tomcat.

Frequently this problem also implies that you should have more web server threads in addition to more database connections. To increase that, edit the `maxThreads` property from this XML file:

`installation-directory/tomcat/conf/server.xml`

© 2003-2011 SmartBear Software

The Server Log
PeerReview Complete is configured by default to perform a very limited amount of logging. The primary purpose of the logs is to record information about error conditions that could potentially arise. When working with SmartBear Technical Support to diagnose issues often it will be helpful to send the logs (or a portion thereof) for investigation.

The logs are located in the following directory:

`installation-directory/tomcat/logs`

Enable server-side debug logging
To enable verbose server-side logging, edit this file:

<installation-dir>/tomcat/webapps/ROOT/WEB-INF/classes/log4j.properties

At the bottom you'll find a line called:

log4j.logger.com.smartbear=info

Change the "info" to "debug." You will have to restart the server for changes to take effect. Remember to revert these changes after reproducing the problem and copying off the log file because this creates a large number of logging messages and degrades server performance.

Change the settings so the server can be accessed remotely
To change the Windows Firewall settings:

Start > All Programs > Accessories > System Tools > Security Center

Click the "Manage security settings for: Windows Firewall" link. On the General tab, make sure the "Don't allow exceptions" box is not checked. On the Exceptions tab, click "Add Port...". For "Name", use "PeerReview Complete", for "Port number", 8080. Leave TCP selected. If you want, you can click the "Change scope..." button to get more options about who can and cannot connect to the port. Click "OK" in each dialog to save the settings and try connecting again.

Contacting Technical Support
For technical support and general inquires:

support@smartbear.com

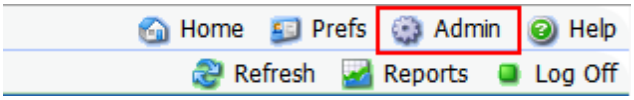
512.257.1569 (M-F, 9am-5pm Central Standard Time)

2.3 PeerReview Complete Configuration

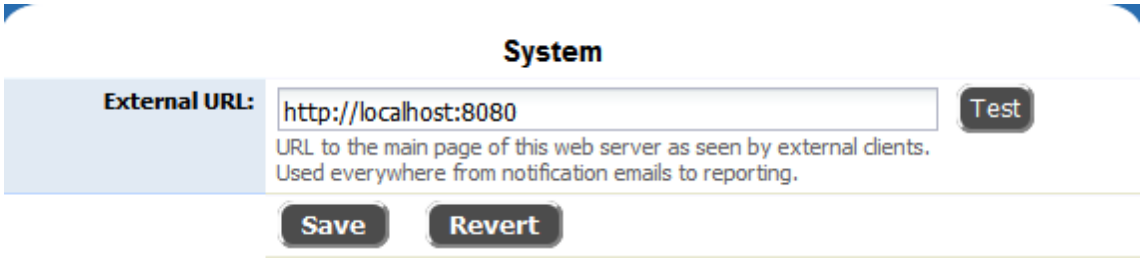
As a system administrator, you can configure PeerReview Complete's settings to however you'd like. This may be useful in better managing users and how they perform reviews.

These settings can be found and changed by simply clicking on "Admin" at the top right menu bar. This button is only available to users that are PeerReview Complete administrators.

© 2003-2011 SmartBear Software

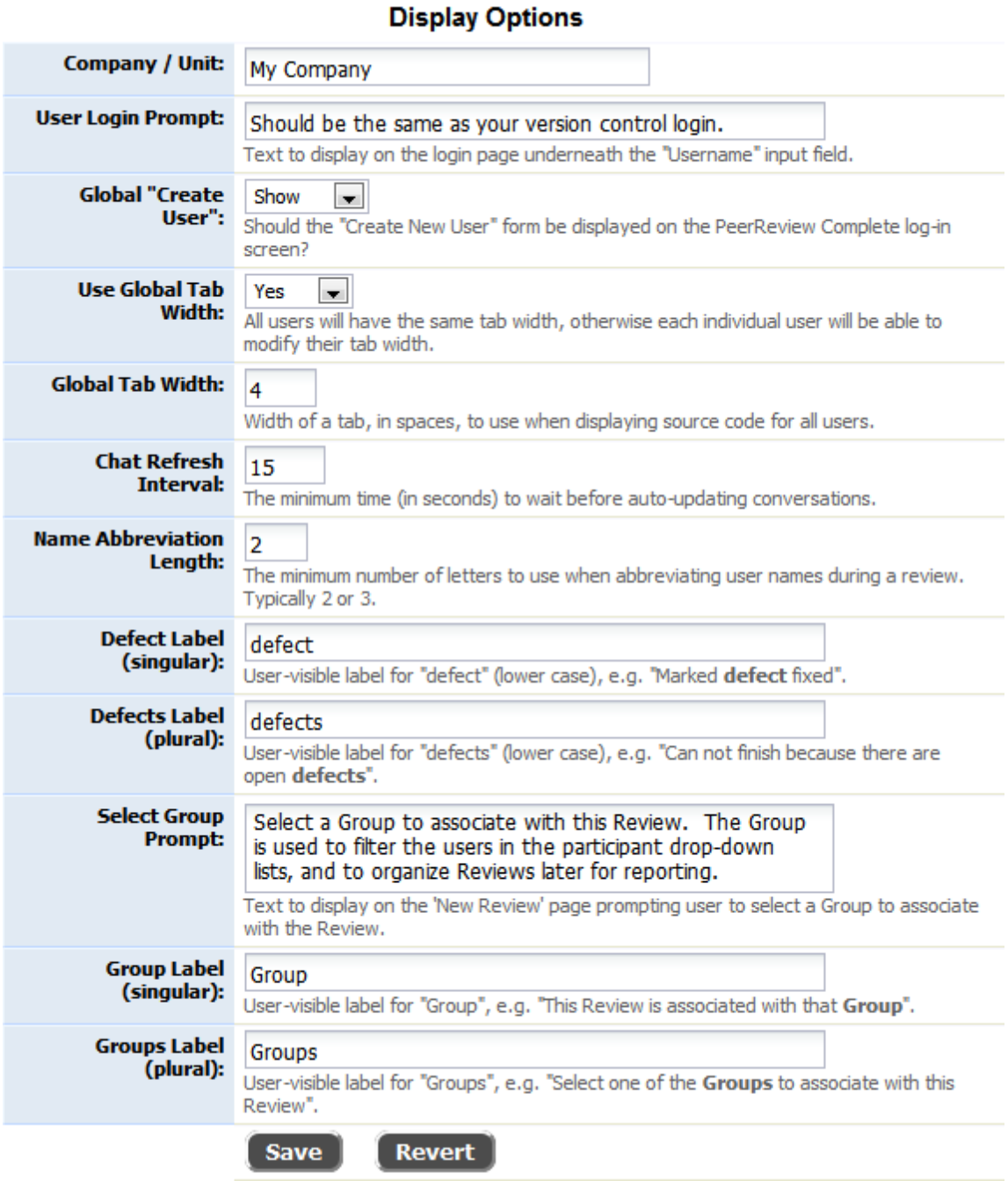


2.3.1 General Settings



External URL The URL to use as the "official" URL for the PeerReview Complete server. This URL will be reported to clients, used in notification emails, and so forth. It should be something that is accessible from as many other computers as possible. It needs to include the port number if it's not `80`.

© 2003-2011 SmartBear Software


Display Options
Company / Unit:
My Company
User Login Prompt:
Should be the same as your version control login.
Text to display on the login page underneath the "Username" input field.
Global "Create User":
Show
Should the "Create New User" form be displayed on the PeerReview Complete log-in screen?
Use Global Tab Width:
Yes
All users will have the same tab width, otherwise each individual user will be able to modify their tab width.
Global Tab Width:
4
Width of a tab, in spaces, to use when displaying source code for all users.
Chat Refresh Interval:
15
The minimum time (in seconds) to wait before auto-updating conversations.
Name Abbreviation Length:
2
The minimum number of letters to use when abbreviating user names during a review. Typically 2 or 3.
Defect Label (singular):
defect
User-visible label for "defect" (lower case), e.g. "Marked defect fixed".
Defects Label (plural):
defects
User-visible label for "defects" (lower case), e.g. "Can not finish because there are open defects".
Select Group Prompt:
Select a Group to associate with this Review. The Group is used to filter the users in the participant drop-down lists, and to organize Reviews later for reporting.
Text to display on the 'New Review' page prompting user to select a Group to associate with the Review.
Group Label (singular):
Group
User-visible label for "Group", e.g. "This Review is associated with that Group".
Groups Label (plural):
Groups
User-visible label for "Groups", e.g. "Select one of the Groups to associate with this Review".
Save
Revert

Company / Unit Arbitrary text that will be displayed at the top of the PeerReview Complete web page. This distinguishes the server from other servers. This personalizes the server and also makes it easy for users to distinguish between different installations you might have.

© 2003-2011 SmartBear Software

User Login Prompt

The text to display on the front page when the user logs in. This text is displayed just below the "Username" field.

The default text suggests using the same login as the version control system, however you might want to change this to be more specific, or in the case of LDAP authentication 50, you could instruct the user to use that login.

Global "Create User"

If you're not using LDAP authentication 50, every user in the system must be explicitly created. The system administrator can create users at any time, but an easier technique is to allow users to create their own accounts from the "Login" screen 124.

When this option is enabled, this create-account form is displayed, otherwise it's hidden.

This option doesn't make sense if you're using LDAP authentication because in that case user accounts are created automatically when a user first successfully authenticates on the Login screen.

Use Global Tab Width

If 'Yes' all users will use the same tab width. If 'No' each user can configure their own preference (the global setting is the default).

Global-Tab Width

Width of a tab (in spaces) for all users to use when displaying reviewed files 160.

Chat Refresh Interval

The minimum time (in seconds) to wait before auto-updating chat conversations 168 in the diff viewer 157.

If increase the refresh time, new chat display will become less responsive, but server load will decrease. This also reduces contention for connections on the browser side because most modern browsers limit the number of simultaneous server connections to 2.

Name Abbreviation Length

The minimum number of letters to use when abbreviating user names during a review.

Defect Label (singular)

User-visible label for "defect" (lower case), e.g. "Marked defect fixed". If the word 'defect' has a negative connotation in your environment, specify a different word (e.g. "item" or "finding," etc.).

Defects Label (plural)

User-visible label for "defects" (lower case), e.g. "Can not finish because there are open defects". If the word 'defects' has a negative connotation in your environment, specify a different word (e.g. "items" or "findings," etc.).

Select Group Prompt

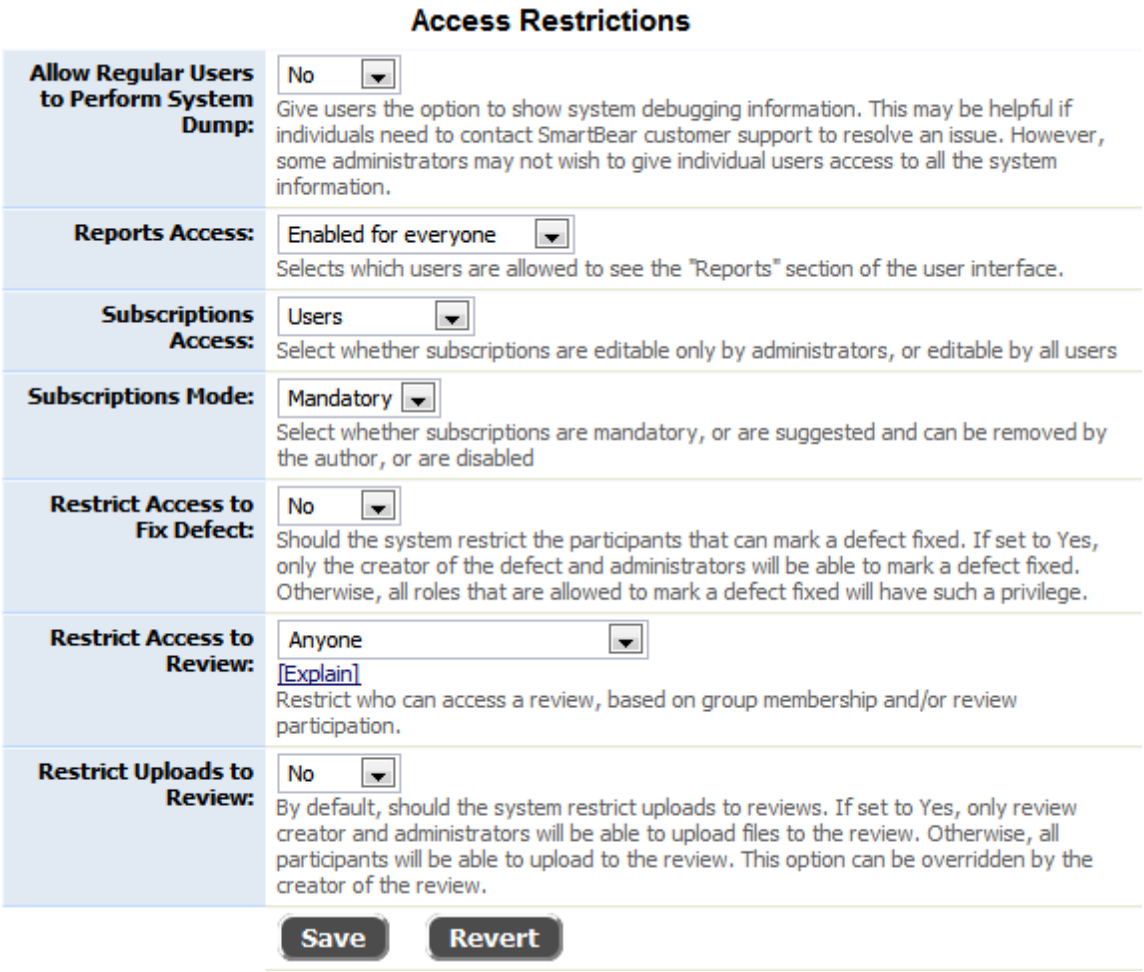
Text to display on the 'New Review' page prompting user to select a Group to associate with the Review.

Group Label (singular)

User-visible label for "Group", e.g. "This Review is associated with that Group".

© 2003-2011 SmartBear Software

Group Label (plural) User-visible label for "Groups", e.g. "Select one of the Groups to associate with the Review".


Access Restrictions
Allow Regular Users to Perform System Dump:
No
Give users the option to show system debugging information. This may be helpful if individuals need to contact SmartBear customer support to resolve an issue. However, some administrators may not wish to give individual users access to all the system information.
Reports Access:
Enabled for everyone
Selects which users are allowed to see the "Reports" section of the user interface.
Subscriptions Access:
Users
Select whether subscriptions are editable only by administrators, or editable by all users
Subscriptions Mode:
Mandatory
Select whether subscriptions are mandatory, or are suggested and can be removed by the author, or are disabled
Restrict Access to Fix Defect:
No
Should the system restrict the participants that can mark a defect fixed. If set to Yes, only the creator of the defect and administrators will be able to mark a defect fixed. Otherwise, all roles that are allowed to mark a defect fixed will have such a privilege.
Restrict Access to Review:
Anyone
[Explain]
Restrict who can access a review, based on group membership and/or review participation.
Restrict Uploads to Review:
No
By default, should the system restrict uploads to reviews. If set to Yes, only review creator and administrators will be able to upload files to the review. Otherwise, all participants will be able to upload to the review. This option can be overridden by the creator of the review.
Save
Revert

Allow Regular Users to Perform System Dump Should regular users be allowed to perform system dump?

Selecting "Yes" will give users, as well as administrators, access to the system debugging information.

Reports Access Who should be allowed to view the "Reports" section of the user interface? You can choose everyone, everyone but with review permissions applied, only administrators, or disabled (no one).

© 2003-2011 SmartBear Software

	Enabled for everyone: Everyone will see the "Reports" section; no restrictions are enforced on what information is returned. Respect Permissions: Everyone will see the "Reports" section; users can only see report content for reviews that they can access. Note that since administrators can always access all reviews, this setting does not affect the content of their reports. Administrators only: Only administrators will see the "Reports" section. Disabled: No one will see the "Reports" section.
Subscriptions Access	Selecting "Users" will allow users, as well as administrators, access to edit subscriptions. Choose "Administrators" to give access to only administrators.
Subscriptions Mode	Mandatory: Subscribed users must be participants in a review. Requested: Subscribed users are automatically added to a review but can be manually removed. Disabled: Subscriptions are not used.
Restrict Access to Fix Defect	Should the system restrict the participants that can mark a defect fixed? If set to Yes, only the creator of the defect and administrators will be able to mark a defect fixed. Otherwise, all roles that are allowed to mark a defect fixed will have such a privilege.
Restrict Access to Review Content	Should the system always restrict access to review content such that only certain users are allowed to view the review? In each review 142 the review creator can elect to further restrict access. Administrators can always access all reviews. Anyone: No global restrictions on who can access reviews Group Based: Users must have access to the review's associated group 88 to access reviews Participants: Users must be participants in the review to access reviews Participants and Group Based: Users must be participants in the review*and* currently have access to the review's associated group to access reviews
Restrict Uploads to Review	Should only creators of a review be allowed to upload files to a review? If set to "yes," only review creators and administrators will be allowed to upload files to the review. However, the option will be available on theCreate Review screen 140 to override at the review creator's discretion. If set to "no," all participants will be allowed to upload files.

© 2003-2011 SmartBear Software

Review Process Options

Allow System Administrator to Perform Reviews:	No Should the System Administrator account be allowed to create reviews? Normally the System Administrator account is reserved for managing licenses and configuring the server. However, in some cases, notably when LDAP authentication is used, the System Administrator may also need to create reviews
Allow "Mark All Read":	Yes Should a user be able to "mark all comments read" without having to click on each one individually in the side-by-side view? This feature makes this operation faster but you run the risk of users not really reading all comments before marking them as "read."
Allow Reopening Completed Reviews:	Don't Allow Once a review has completed, should participants be allowed to re-open the review just by making an additional comment? If not allowed, an administrator is still able to re-open a review by clicking a link.
Allow Editing General Information of Completed Reviews:	Don't Allow Once a review has completed, should participants be allowed to edit the review title and custom fields? This setting does not apply to participant custom fields.
Create a Commit Action Item for Completed Reviews:	Create When a pre-commit review has completed, should the review author(s) get an Action Item prompting to commit the changes?
Allow Deleting/Cancelling Reviews:	Allow Allow the review creator or author to cancel the review at any time, otherwise only an administrator can cancel a review. Other participants are never allowed to cancel a review.
Default Review Deadline:	0 days If set to non-zero, this is the default number of days until reviews are due. If zero, users will not be prompted for a review deadline.
Show Metrics:	Yes Should users be able to see that metrics are collected including time spent in review, number of defects, and so on? If disabled, metrics are still collected and can be retrieved from the database, however the information is not displayed to the user during review or in the reports section.
Allow Combined Review View:	No Give users the option to display multiple changelists as a single changelist during the review. We don't recommend this because the combined "changelist" can be confusing to understand when the multiple changelists are not sequential.
Character Sets for Diff Viewer:	UTF-8 UTF-16LE UTF-16BE ISO-8859-1 ISO-8859-15 Give users the option to view file contents in various character sets. Usually PeerReview Complete's character set auto-detection will detect the correct character set, but when it does not, users will be able to select between these character sets for displaying the code. To avoid confusion, limit this list to the character sets that you expect your users to require. [Supported Character Sets]
	Save Revert

© 2003-2011 SmartBear Software

Allow System Administrator to Perform Reviews

Should the main system administrator be allowed to participate in reviews?

If you're using internal authentication the setting is typically "no" because the main "admin" account is special and should be used only for system configuration and not to actually do reviews.

If you're using LDAP authentication [50] the setting is typically "yes" because the main administrator is usually an actual human being who will also want to participate in reviews.

This setting affects whether the system account is allowed to create a new review [140] and whether it appears in e.g. drop-down lists for review participants [142].

Allow "Mark All Read"

Should users be given the option to "mark all conversations read" in one click in the conversation area of the diff viewer [167]?

This is a convenient operation, so most administrators leave this enabled. However this makes it easy for someone to not actually read a lot of comments because they are not forced to visit each conversation individually.

Allow Reopening Completed Reviews

Once a review has completed, should participants be allowed to re-open the review just by making an additional comment?

If not allowed, an administrator is still able to re-open a review by clicking a link.

Allow Editing Completed Reviews

Once a review has completed, should participants be allowed to edit the review title and custom fields?

Create a Commit Action Item for Completed Reviews

By default, when a pre-commit review is completed, the author(s) are given an action item as a reminder to commit the changes to the underlying source control system. The author then has the option of using the Tray Notifier to commit the changes. In some workflows, this is not the correct behavior. In that case, choose "Do Not Create" to avoid this prompt.

Allow Deleting/ Cancelling Reviews

Should the review creator or author be able to cancel the review at any time?

If "Don't allow" is selected, only an administrator will have the permissions to cancel a review. Other participants are never allowed to cancel a review.

Default Review Deadline

If set to non-zero, this is the default number of days until reviews are due. If zero, users will not be prompted for a review deadline [141] when creating a new review.

© 2003-2011 SmartBear Software

Show Metrics

Should users be able to see that metrics are collected including time spent in review, number of defects, and so on?

If set to "No", metrics are still collected and can be retrieved directly from the database 427, however the information is not displayed to the user during review 156 or in the reports 176 section.

Allow Combined Review View

Allow the review creator to specify that multiple changelists should be displayed combined as if they were one large changelist.

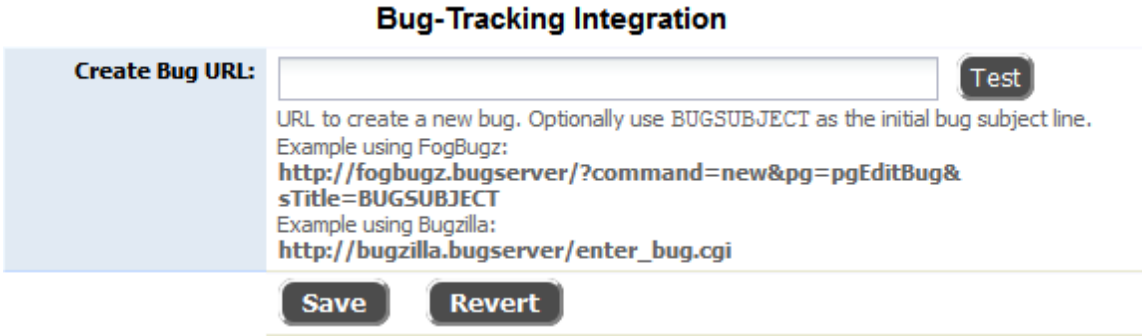
We do not recommend using this setting unless you're sure your users will understand what they are seeing. This mode can confuse changes that are actually part of the review with changes that happened between files in the changelist.

Character Sets for Diff Viewer

Give users the option to view file contents in various character sets.

Usually PeerReview Complete's character set auto-detection will detect the correct character set, but when it does not, users will be able to select between these character sets for displaying the file contents. To avoid confusion, limit this list to the character sets that you expect your users to require.

Click `[Supported Character Sets]` to view all supported character sets.



Create Bug URL

The URL to your bug tracking system that creates a new bug. Optionally use the special text `BUGSUBJECT` as the starting subject line for the bug. This setting is used when the user is prompted to create a bug in your external issue tracking system.

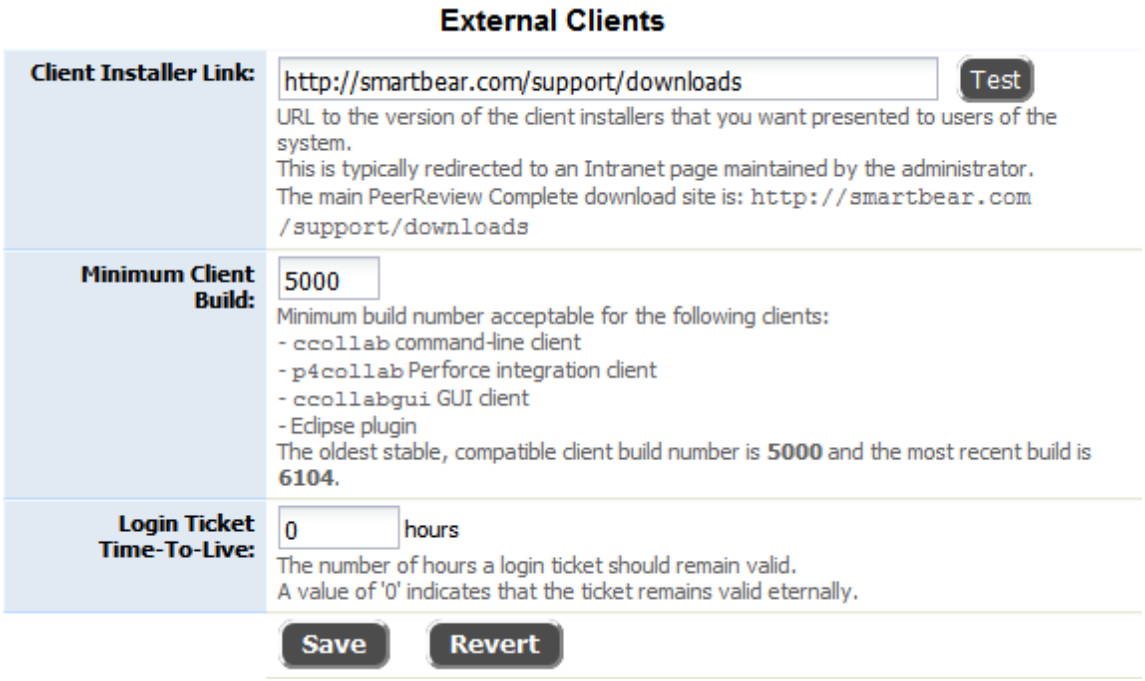
For example, with FogBugz the URL might look like this:

`http://bugserver/default.php?command=new&pg=pgEditBug`

In Bugzilla the URL would look something like this:

`http://bugserver/enter_bug.cgi`

© 2003-2011 SmartBear Software


External Clients
Client Installer Link:
http://smartbear.com/support/downloads
Test
URL to the version of the client installers that you want presented to users of the system.
This is typically redirected to an Intranet page maintained by the administrator.
The main PeerReview Complete download site is: http://smartbear.com/support/downloads
Minimum Client Build:
5000
Minimum build number acceptable for the following clients:
- ccollab command-line client
- p4collab Perforce integration client
- ccollabgui GUI client
- Eclipse plugin
The oldest stable, compatible client build number is 5000 and the most recent build is 6104.
Login Ticket Time-To-Live:
0
hours
The number of hours a login ticket should remain valid.
A value of '0' indicates that the ticket remains valid eternally.
Save
Revert

Client Installer Link

URL to the version of the client installers that you want presented to users of the system.

This is typically redirected to an intranet page maintained by the administrator.

Minimum Client Build

Minimum allowable build number for the various client applications including the Command Line Client [196], the GUI Client [189], and the Tray Notifier [278].

Use this to ensure your clients are reasonably up to date. This is especially important if there is a feature or bug-fix you know is necessary for your system.

The help text in the GUI will identify the oldest stable, compatible build number and will list the most recent known build number for your reference.

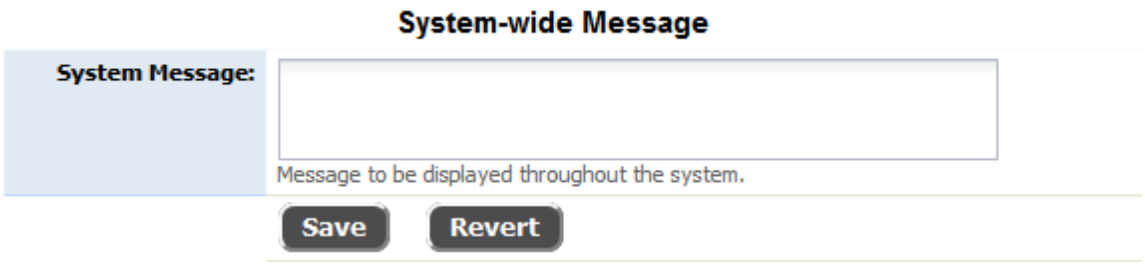
Minimum Legacy Windows GUI Client Build

Minimum allowable build number for the Legacy Windows GUI Client. This client has been deprecated and is no longer supported but is still in use at some customer sites.

Use this to ensure your clients are reasonably up to date. This is especially important if there is a feature or bug-fix you know is necessary for your system.

© 2003-2011 SmartBear Software

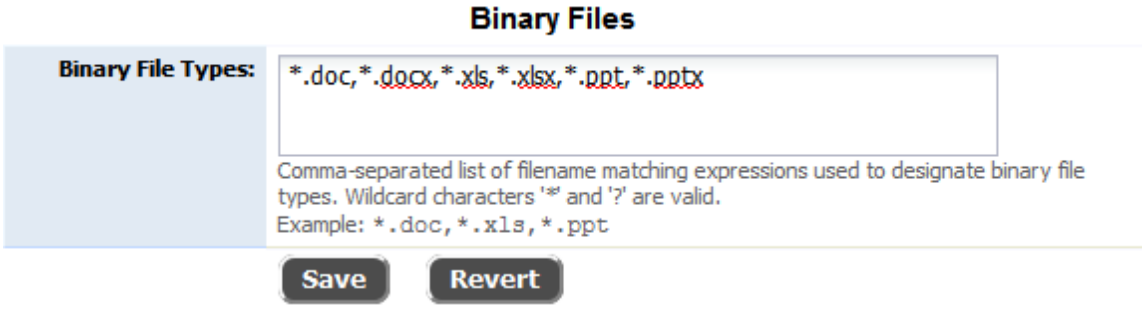
The help text in the GUI will identify the oldest stable, compatible build number and will list the most recent known build number for your reference.



System Message

Administrators can optionally insert a message to be displayed throughout the system.

This can be useful for reminders, PeerReview Complete announcements, or general information.


Binary Files
Binary File Types:
.doc,.docx,*.xls,*.xlsx,*.ppt,*.pptx
Comma-separated list of filename matching expressions used to designate binary file types. Wildcard characters '*' and '?' are valid.
Example: *.doc,*.xls,*.ppt
Save
Revert

Binary File Types

Binary files attached to reviews are not displayed in the Diff Viewer 157, and instead must be opened by external applications. Here you can configure which files are to be treated as binary. By default, Microsoft Word, Microsoft Excel and Rich Text files are considered binary.

Filename matching is done using the '*' and '?' wildcards characters. '*' matches 0 or more contiguous characters, and '?' matches exactly one character. The character matching is case-insensitive on Windows platforms and case-sensitive on all other platforms.

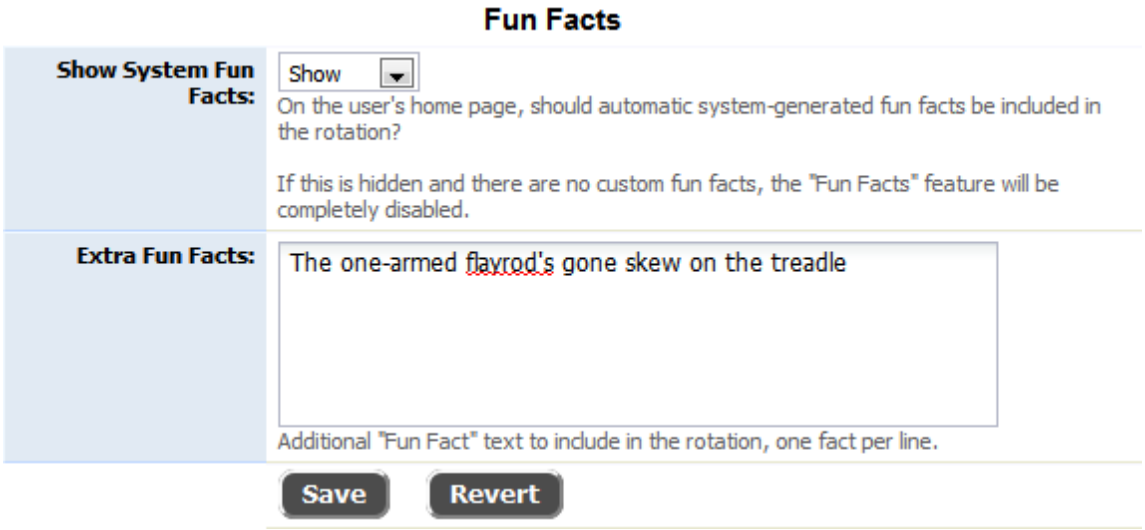
© 2003-2011 SmartBear Software

Image Files
Image File Types:
.gif,.jpg,*.jpeg,*.png
Comma-separated list of filename matching expressions used to designate image file types. Wildcard characters '*' and '?' are valid.
Example: *.gif,*.png,*.jpg
Save
Revert

Image File Types

Image files are reviewed in the browser in the diff viewer using a special image diff viewer [162]. Here you can configure which files are treated as images. By default, GIF, JPEG, and PNG images are handled as images. The content of the files must be renderable by the users' browsers for images to actually be reviewable. Images that require special software to view should be treated as Binary Files [82] so they can be rendered by that software.

Filename matching is done using the '*' and '?' wildcards characters. '*' matches 0 or more contiguous characters, and '?' matches exactly one character. The character matching is case-insensitive on Windows platforms and case-sensitive on all other platforms.


Fun Facts
Show System Fun Facts:
Show
On the user's home page, should automatic system-generated fun facts be included in the rotation?
If this is hidden and there are no custom fun facts, the "Fun Facts" feature will be completely disabled.
Extra Fun Facts:
The one-armed flayrod's gone skew on the treadle
Additional "Fun Fact" text to include in the rotation, one fact per line.
Save
Revert

Show System Fun Facts

Fun facts are lines of information displayed on the home page. PeerReview Complete rotates through a number of internal facts based on aggregate review metrics.

If this is set to "Hide" and there are no extra fun facts listed, this feature is completely disabled.

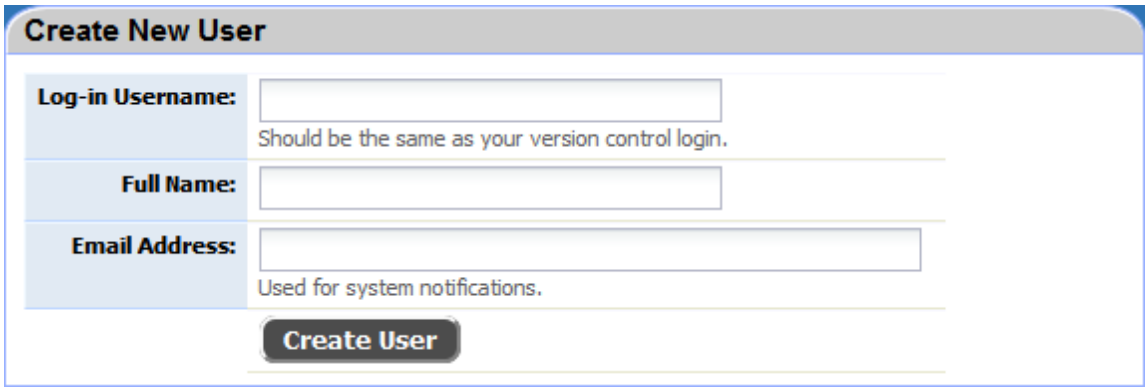
© 2003-2011 SmartBear Software

Extra Fun Facts Additional lines to be displayed can be entered in this field.

2.3.2 User Management

The user management page lets you add, disable, and manage user accounts.

Creating New Users
This form lets you create new users in PeerReview Complete. If you're using LDAP authentication you won't see this form because users are created automatically when they first log into the system.



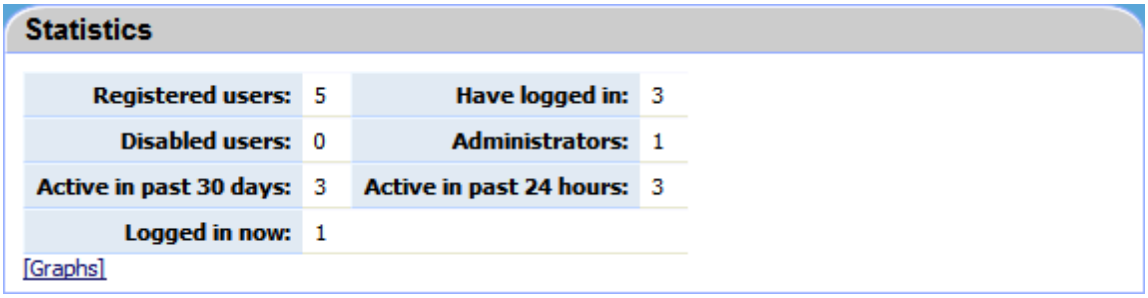
Login names must be unique in the system; PeerReview Complete will not allow you to create a second user with the same login. The full name is used when displaying the user in the interface, and the email address is used to send notification messages.

After you create a user you are returned to the same screen. This makes it easy to add multiple users quickly.

> **Warning:** When creating new user accounts, it's a good idea to make the login name identical to the user's version control login. One advantage is that the user doesn't have to remember a new login. Another advantage is that certain features are enabled in PeerReview Complete because it can correlate users from version control changelists with PeerReview Complete users. Finally cross-system reporting becomes much easier when the logins are the same.

Statistics
The statistics box shows information about users and usage of PeerReview Complete:

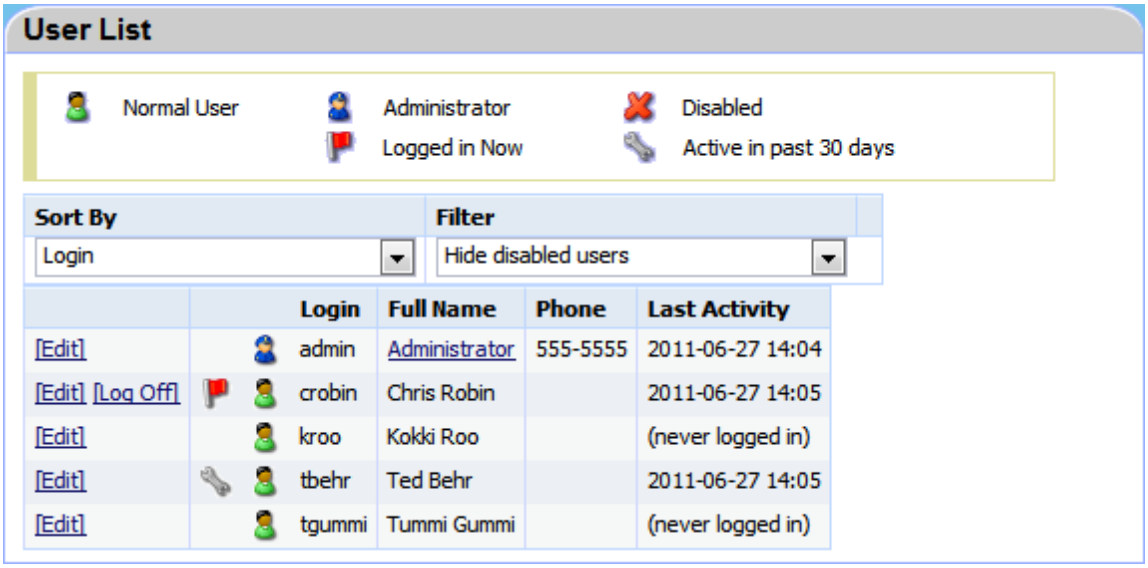
© 2003-2011 SmartBear Software



This can help monitor your usage and determine how many seats are necessary.

The User List

The user list shows every user in the system along with key information about each:



You can sort the user list by:

- Login
- Full Name
- Last Activity (oldest first)
- Last Activity (most recent first)

You can filter the user list by:

- Show all users
- Hide disabled users
- Show only logged-in users
- Show only administrators
- Show users missing name
- Show users missing email
- Show users missing phone number

© 2003-2011 SmartBear Software

The [Edit] link allows you to update any user setting 127 from basic contact information to personal display and notification settings. The [Log Off] link allows you to immediately log off a logged-in user 86.

The first column of icons tells you several things about the user's recent activity and how the user is counting towards server licensing 41.

Icon	Logged in?	Consuming floating-seat license?	Consuming fixed-seat license?
	Yes	Yes	Yes
	No	No	Yes
(none)	No	No	No

The second column of icons tells you about the user's access rights:

Icon	Is user an administrator?	Is user allowed to log in?
	Yes	Yes
	No	Yes
	No	No

You will notice that the admin account never counts towards licensing. This is one of the special characteristics of the System Administrator Account 87.

Log User Off

Administrators also have the ability to force a user to log out of PeerReview Complete to make a seat available. To do so, just click the [Log Off] link that will appear to the left of the logged in username. This feature should be used carefully as the user will immediately be logged off. Once a user is forced out, the next time they access PeerReview Complete, they will be required to reenter their username/password to gain access.

Deleting & Disabling Users

You cannot delete users in PeerReview Complete because the user record is necessary for reporting and for viewing old reviews that they were a part of.

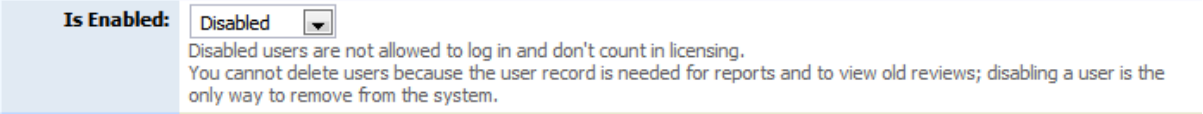
However you can *disable* a user. Disabling a user does several things immediately:

- The user can no longer log in.
- If the user is already logged in, he is automatically and immediately logged off.
- The user doesn't show up in standard user lists for e.g. inviting to a review.
- The user will not receive any notifications or broadcast messages 99.
- The user immediately does not count towards either floating-seat or fixed-seat licensing 41.

The user will continue to appear in the general user list at the end, and an administrator can re-enable a user at any time.

© 2003-2011 SmartBear Software

To disable or enable a user, click the `[Edit]` link next to that user on the user list and change the "Is Enabled" flag:

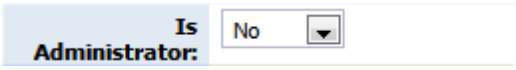

Is Enabled:
Disabled
Disabled users are not allowed to log in and don't count in licensing.
You cannot delete users because the user record is needed for reports and to view old reviews; disabling a user is the only way to remove from the system.

Administrative Access

Users with administrative access are allowed to access the "Admin" menu from the menubar and all screens therein. They also have a variety of other permissions described elsewhere in this manual, including but not limited to:

- Administrators can look at *all* reviews, regardless of whether they have been marked restricted 140.
- Administrators can do anything with defects in *any* review: create, delete, edit, externalize, and mark fixed, regardless of their role or even non-participation in that review.
- Administrators can edit characteristics of *any* review including changing custom field content, changing the list of participants, and canceling the review.

To set or clear a user's administrative access, click the `[Edit]` link next to that user on the user list and change the "Is Administrator" flag:



Warning: Be careful if you are an administrator and you're editing your own user information. If you say that you're not an administrator, you're not from that point forward!
If you do this by accident it can be remedied by another administrator, or the System Administrator in particular (see below).

The Special "System Administrator" Account

One user account is special: The System Administrator Account. How do you know which account is this special one? If you're using internal authentication, the System Administrator Account is always called `admin` (this cannot be changed). If you're using LDAP authentication 50, the account is determined during installation 31.

The System Administrator Account has the following special properties:

- There can be only one. One to rule them all.
- The System Administrator is *always* an administrator; administrative access cannot be revoked. (This is prevented in the user interface so you can't revoke access, even by mistake.)
- The System Administrator can *always* log in. Even if the server license has been exceeded, this account is *still* allowed to log in.

© 2003-2011 SmartBear Software

2.3.3 Groups

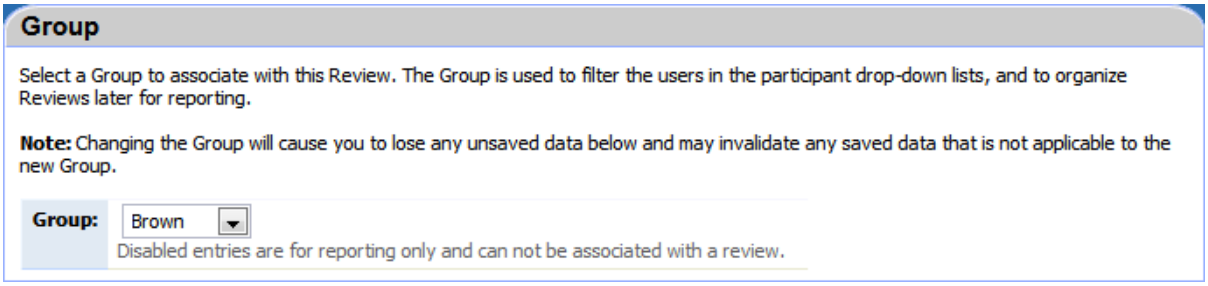
Group Administration

User "Groups" is an optional feature that allows you to define groups of users within your organization, so reviews can be associated with a group. Groups can be used to model your organization's hierarchy and/or project assignments. Each group can contain multiple users and can also contain other groups. Each user can be in more than one group. You can specify group administrators who are able to maintain individual groups.

The word "Group" is configurable in the General Settings 73, so you could change it to "Team," or "Project," etc.

Association With Reviews

If a user is a member of more than one group that can be associated with a review 91, an additional field is displayed when creating a review:


Group
Select a Group to associate with this Review. The Group is used to filter the users in the participant drop-down lists, and to organize Reviews later for reporting.
Note: Changing the Group will cause you to lose any unsaved data below and may invalidate any saved data that is not applicable to the new Group.
Group:
Brown
Disabled entries are for reporting only and can not be associated with a review.

The list of groups to select from is calculated by first creating a list of all groups that contain the review creator as a member. All groups in that list are then checked to see if they are a child group of any other group(s). Parent groups are added to the list recursively.

In the settings for a group there is an option to disallow associating that group with reviews 91. This is useful for creating groups that are used for reporting or filtering only. Groups that cannot be associated with a review are still shown in the list, but cannot be selected.

Associating a group with a review makes it easier to select participants for that review because the Person list can optionally be filtered by group. The filter is calculated by starting with the users that are members of the selected group. If the selected group also contains child groups, then the users that are members of those child groups will also be added. The search for child groups is done recursively.

Restricting Access

Groups can be used to limit access to reviews by setting **Restrict Access to Review Content** 76 in the general settings. When the group access restriction feature is used, the list of users who have access to the review is calculated by starting with the users that are members of the group that is associated with the review. If that group also contains child groups, then the users that are members of those child groups will also be added. The search for child groups is done recursively. Finally, if the group associated with the review is a child of any other group(s) then the users that are members of those parent groups will also be added. The search for parent groups is done recursively.

Sample Group Hierarchies

It may be difficult to understand how best to use groups based on the rules given above. We've provided a few examples of how they are usually used:

© 2003-2011 SmartBear Software

Using Groups for Organizational Hierarchy 94
Using Groups for Projects 95
Using Groups for Organizational Hierarchy and Projects 97

Maintaining Groups

Groups can be created and modified in one of two ways: manually (with the web user interface - see the sections below, starting with **Creating New Groups** 89, or with the command-line scripting command ccollab admin group create 272) or automatically via the command-line scripting ccollab admin group sync 274 command.

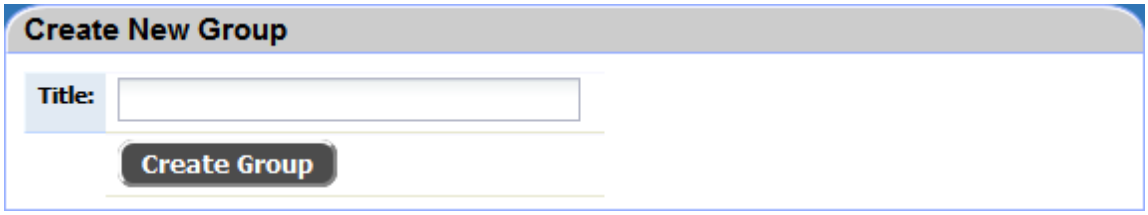
"Group Administrators" can be assigned to administer individual groups. A group administrator is a regular (not full administrator) user who can manage a specific group. Group administrators are allowed to create child groups of groups they administer, but are not allowed to create top-level groups.

If you have only a few groups it's easiest to manage them manually. If you have many groups or if you're mirroring them from an external system like LDAP, it's best to manage them automatically using the "sync" command.

The command-line "sync" command interface is provided to make it easy to mirror group definitions in to PeerReview Complete from an external system. Theccollab admin group sync 274 command uses the contents of an XML file (schema) to update PeerReview Complete's group definitions. See theSyncing Groups 92 topic for more information.

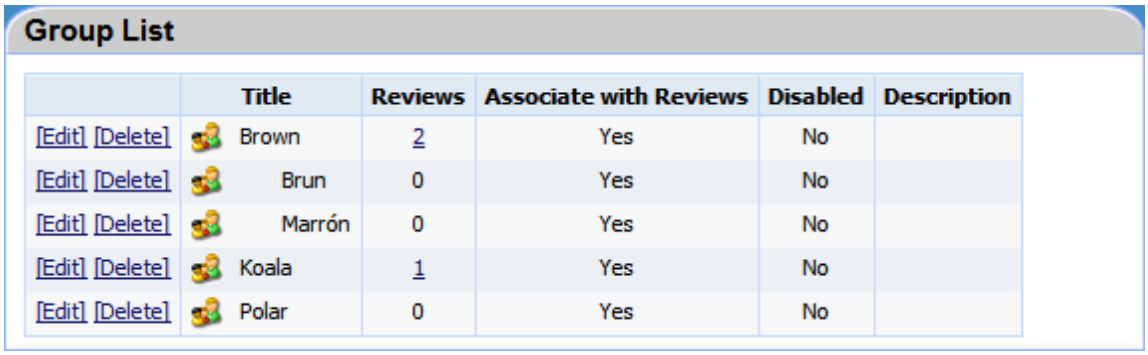
PeerReview Complete tracks which groups were created manually and which were defined viaccollab admin group sync 274. Groups that were defined manually are not overwritten as a result of usingccollab admin group sync 274. See the Using Groups for Organizational Hierarchy and Projects 97 example to see how this can be useful.

Creating New Groups



To make use of this feature, first create a title for your group, and click "Create Group."

Group List



© 2003-2011 SmartBear Software

Once a group is created, the group will appear in the Group List. In the Group List, you can edit the configuration details or delete the group by clicking the appropriate link. The Group List also shows the number of reviews associated with the group, whether or not the group can be associated with reviews, whether or not the group is enabled, and any descriptions given to the group. Note group administrators will only see groups listed that they can administer.

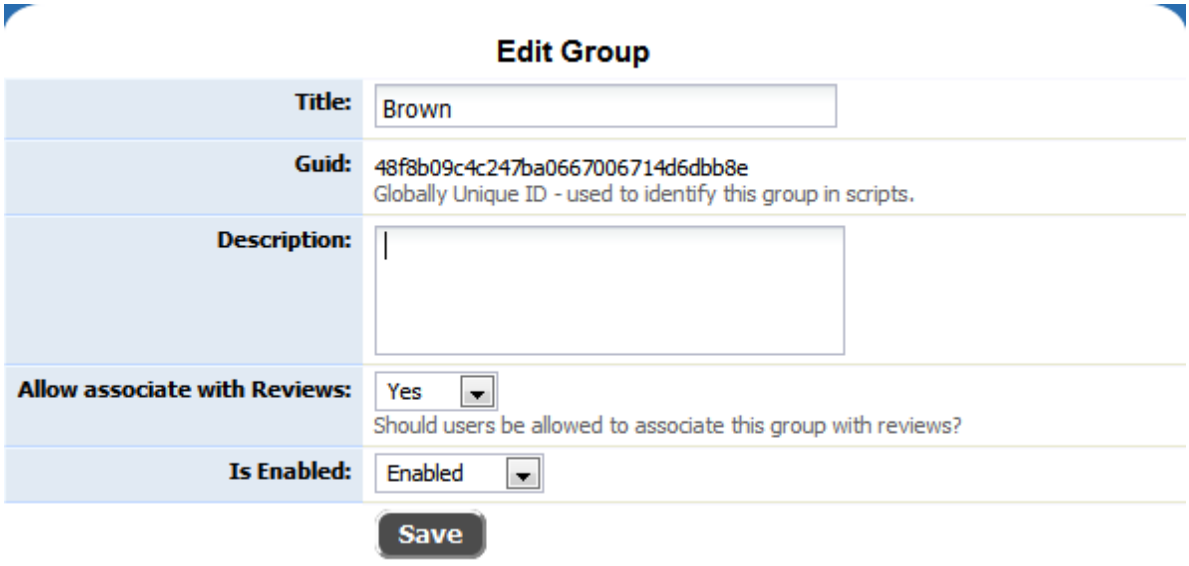
Deleting a Group

To delete a group, click the `[Delete]` link next to the group's title in the Group List. You can only delete a group if it is not associated with reviews. If the group is associated with reviews, you must first associate those reviews with a different group before deleting it; alternatively, you could instead disable the group in order to prevent future reviews from being associated with it.

Group administrators are further restricted to only being able to delete groups that they are group administrators of and that have no child groups. This restriction is to prevent group administrators from indirectly creating top-level groups.

Editing a Group

To edit a group, click on the `[Edit]` link next to the group's title in the Group List. You will be directed to the "Edit Group" page.



Title Human-readable title for the group, used in drop-down menus, etc... This does not have to be unique across all groups.

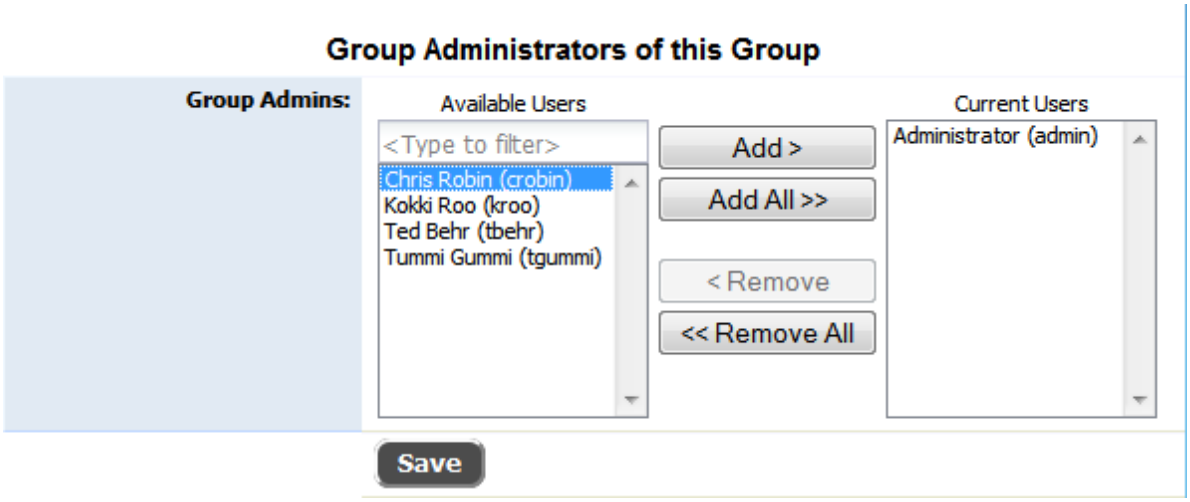
Guid Machine-readable ID for group, unique across all groups. If the group was created using the web interface, this is generated automatically. If the group was created using the sync 274 command, this ID is supplied in the XML file.

Description Human-readable description for the group. This is displayed on the "New Review 140" page when associating a group with a review 88.

© 2003-2011 SmartBear Software

Allow associate with Reviews If this is set to "Yes", this group is selectable from a drop-down list when associating a group with a review 88. If "No", this group may still be displayed in the list, but will be disabled. For an example, see Using Groups for Organizational Hierarchy and Projects 97.

Is Enabled Groups can be disabled when they are no longer in use. Disabled groups still show up in reports because reviews may be associated with them.



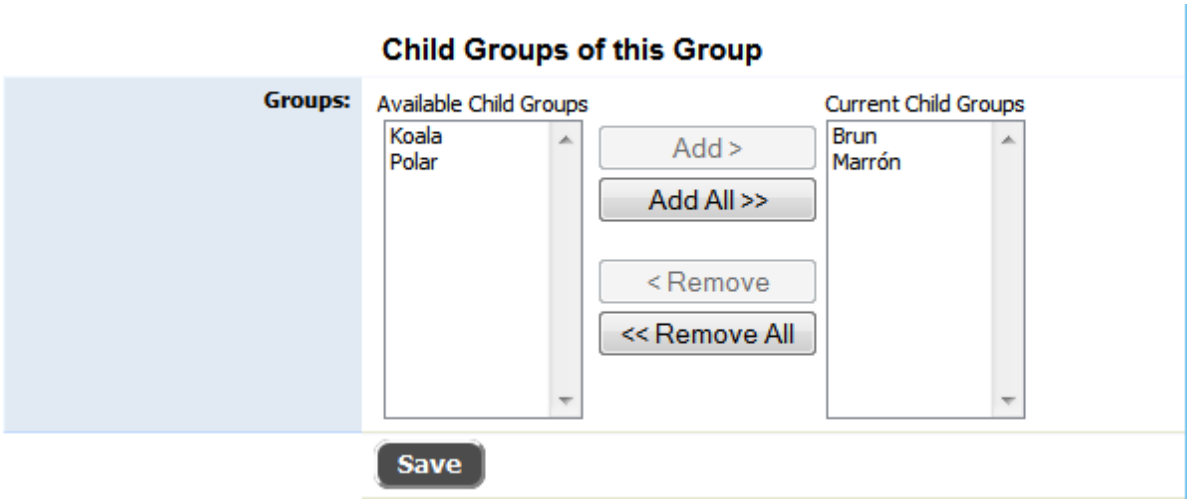
Group administrators are regular (not full administrator) users who are allowed to administer this group. Group administrators can edit 90 all of the fields of the group.

Group administrators can add or remove group administrators for the group, even themselves, but they can not remove the last group administrator from the group.

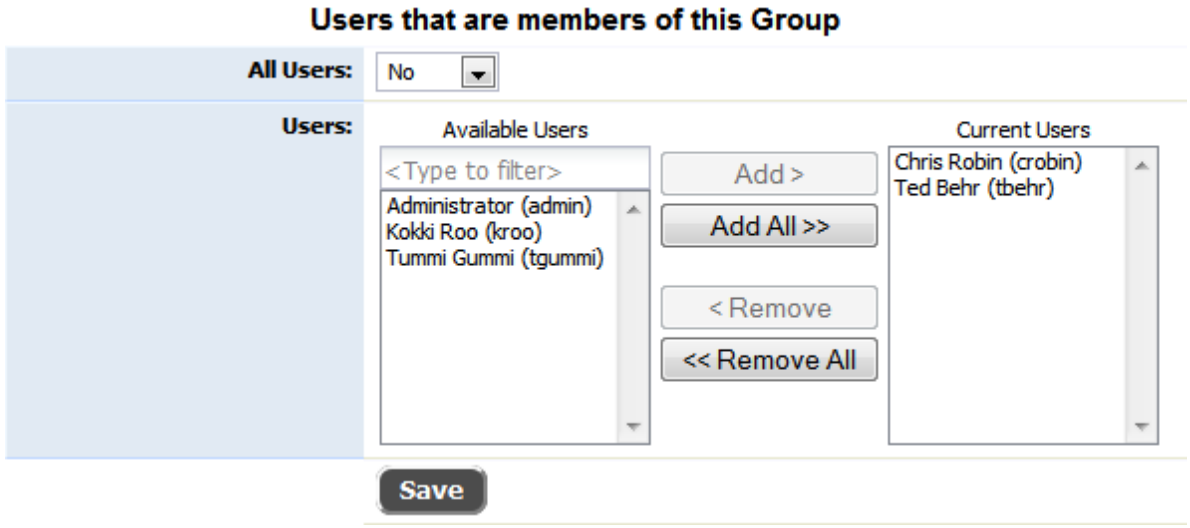
Group administrators can add or remove user members 92 of the group. Users that are group administrators may or may not also be regular members of the group.

Group administrators are not allowed to directly edit the list of child groups of the group, but they can create new child groups of the group. The group administrator that creates a child group is automatically a group administrator of that child group.

© 2003-2011 SmartBear Software



You can add multiple groups to be children of the group. This may be helpful if a group is constituted by a number of sub-groups. The "group hierarchy" created by Group -> Child Group relationships is important for the associate with review 88 and access control 88 algorithms.



Groups can have specifically named member users, or they can automatically contain all users as members. Group membership is important for the associate with review 88 and access control 88 algorithms.

2.3.3.1 Syncing Groups

PeerReview Complete provides a simple web UI for creating and managing groups, but when groups are used to represent organizational hierarchy 94 for large companies the web UI becomes unmanageable. Such groups are usually already stored in an external system such as LDAP, so PeerReview Complete provides the ability to "sync" it's groups from an external system.

© 2003-2011 SmartBear Software

Group Sync XML

The `ccollab admin group sync` [274] command takes as input an XML file describing all the groups being synced, their relationships, and their member users. The XML format is simple and a full schema is provided. For an example see Using Groups for Organizational Hierarchy [94]. The PeerReview Complete administrator should write a script that queries the external system (i.e. LDAP) and creates this XML.

Syncing

The PeerReview Complete administrator should run the sync script periodically. PeerReview Complete will examine the group sync xml and make the appropriate changes in the system. Processing the sync is not very resource intensive, so you probably don't need to worry about load on the PeerReview Complete server.

Users

The group sync xml describes which users are members of which groups. There are two options if the XML describes users who don't yet have accounts in PeerReview Complete, controlled by the --create-user [274] option of the `ccollab admin group sync` [274] command:

- ***Create the member-user*** - If the --create-user [274] option is specified, a user account is created for the user. Note the user does not consume a license [41] until they log in.
- ***Ignore the member-user*** - If the --create-user [274] option is not specified (default behavior), the member-user is ignored and a user account is not created. This is useful if you have a large organization but only a small subset of the employees are using PeerReview Complete.

Removing Groups

Groups may be removed from the external system. There are two options if the XML no longer describes a group which was created in an earlier sync, controlled by the --delete-groups [275] option of the `ccollab admin group sync` [274] command:

- **Delete the group** - If the --delete-groups [275] option is specified, the group is deleted. However if the group can not be deleted because it has associated reviews [90], an error is returned and the sync is canceled.
- **Disable the group** - If the --delete-groups [275] option is not specified (default behavior), the group is marked disabled. This always works, regardless of whether there are reviews associated with the group.

Renaming groups

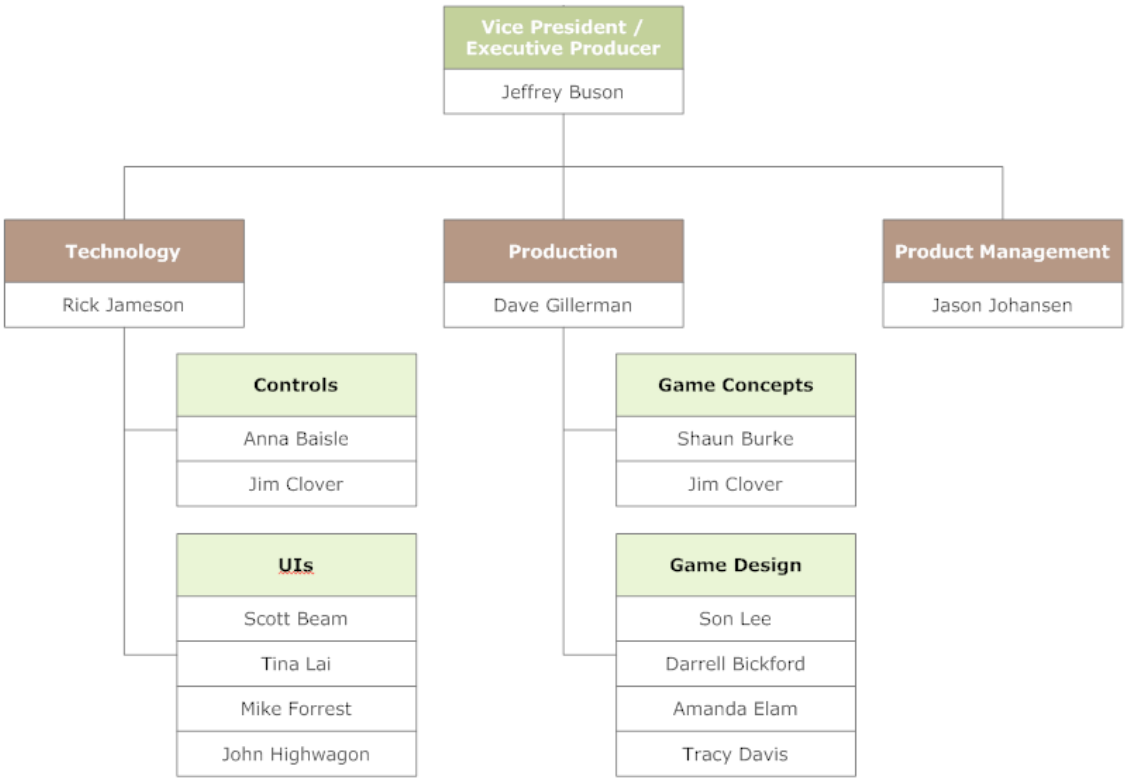
Groups are uniquely identified by a GUID which is specified in the group sync xml. To rename the group simply specify a group with the same GUID but a different title.

© 2003-2011 SmartBear Software

Manual groups
Sometimes it is useful to sync groups with an external system but *also* manually specify some groups using the web interface or the ccollab admin group create 272 command. For an example see Using Groups for Organizational Hierarchy and Projects 97. PeerReview Complete keeps track of which groups were created using sync and which groups were created manually. Manually created groups are not deleted or disabled on subsequent syncs just because they're not in the group XML. The web UI displays a warning when manually editing a group that was created by sync. Manually created groups can contain synced groups as children, but not vice-versa.

2.3.3.2 Using Groups for Organizational Hierarchy

The Groups 88 feature is flexible enough to allow you to model any organization. As an example, suppose your company's **Action Games Division** org chart looked like this:



Companies this large typically store their org chart in an LDAP system. The PeerReview Complete administrator can write a script thatsynchronizes 92 PeerReview Complete groups with the LDAP system.

© 2003-2011 SmartBear Software

For example, the XML to create the organizational structure shown above would look like:Sample XML

```
<groups>-<group guid="action-games-division" title="Action Games Division"
allow-associate-with-reviews="false"><member-group guid="technology"/><member-group
guid="production"/><member-group guid="product-management"/><member-user
login="jbuson"/></group>-<group guid="technology" title="Technology"
allow-associate-with-reviews="false"><member-group guid="controls"/><member-group
guid="uis"/><member-user login="rjameson"/></group>-<group guid="production"
title="Production" allow-associate-with-reviews="false"><member-group
guid="game-concepts"/><member-group guid="game-design"/><member-user
login="dgillerman"/></group>-<group guid="product-management" title="Product
Management" allow-associate-with-reviews="false"><member-user
login="jjohansen"/></group>-<group guid="controls" title="Controls"
allow-associate-with-reviews="false"><member-user login="abaisle"/><member-user
login="jclover"/></group>-<group guid="uis" title="UIs"
allow-associate-with-reviews="false"><member-user login="sbeam"/><member-user
login="tlai"/><member-user login="mforrest"/><member-user
login="jhighwagon"/></group>-<group guid="game-concepts" title="Game Concepts"
allow-associate-with-reviews="false"><member-user login="sburke"/><member-user
login="jclover"/></group>-<group guid="game-design" title="Game Design"
allow-associate-with-reviews="false"><member-user login="slee"/><member-user
login="dbickford"/><member-user login="aelam"/><member-user
login="tdavis"/></group></groups>.
```

Note - we have assumed that user's logins are created by prefixing the first character from their first names to their last names. All of the groups have "Allow associate with Review[91]" set to "No".

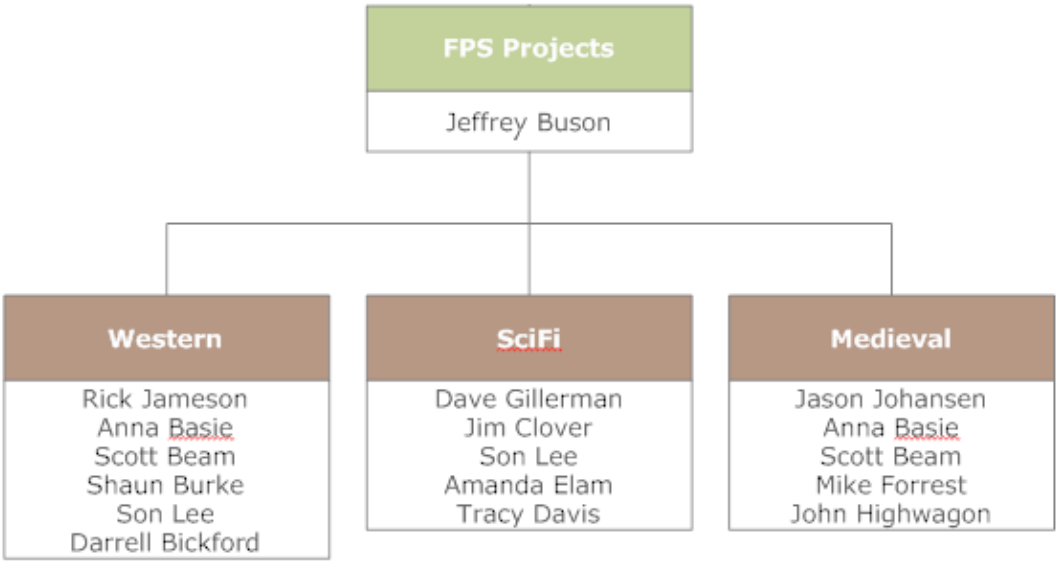
Reporting based on group membership

The reporting system can be used to create reports of e.g. all reviews where a member of a department was a participant, or all reviews created by a member of a department, etc...

2.3.3.3 Using Groups for Projects

The Groups[88] feature is flexible enough to allow you to model any structure. As an example, if you wanted to model a group of projects from a software gaming company:

© 2003-2011 SmartBear Software



Since there's only a few groups we create them manually using the web UI. First create a group called *FPS Projects*. The *FPS Projects* group would have only one user in it: Jeffrey Buson. It would also have three child groups: *Western*, *SciFi*, and *Medieval*. The *Western*, *SciFi*, and *Medieval* groups would each contain the users that are assigned to those projects.

Associating groups with reviews

We set "Allow associate with Review 91" to "Yes" for the *Western*, *SciFi*, and *Medieval* groups. In this example we are using groups to represent projects, so we configure the singular group label 75 to "Project" and the plural group label 76 to "Projects".

When a user in one of those groups creates a review it will be associated with a group. Most of the users are only members of one group, so when they create a review they don't have to manually choose a group - the review is automatically associated with their group. The users "Anna Basie" and "Scott Bean" are members of both the *Western* and *Medieval* groups, so when they create a review they are prompted to select one of those "Projects" to associate with the review.

Filtering the participant list

When a user creates a review they can choose to filter the participant list 88 by that group. When Amanda Elam creates a review, if she selects "Filter possible participants by 'SciFi' Project" then the participant drop-downs will only contain Dave Gillerman, Jim Clover, Son Lee, and Tracy Davis. This is useful at large companies to make the list of users more manageable.

Restricting Access based on group

Access to reviews can be restricted based on associated group. The restriction can be set for all reviews or just for a particular review. For example, if a review was associated with the *Western* group and group access restriction was enabled, then it would be accessible to Scott Beam and Jeffrey Buson, but not accessible to Jason Johansen.

© 2003-2011 SmartBear Software

Reporting based on associated group
You can display, sort and filter reports by the associated group, so you know which project the review was for.

Finishing projects
When a project is finished, we mark it's group disabled. This means the group can no longer be associated with reviews, but it's still in the system for reporting purposes.

2.3.3.4 Using Groups for Organizational Hierarchy and Projects

The Groups 88 feature is flexible enough to allow you to model any structure. You might want to use groups to represent an organizational hierarchy like the Using Groups for Organizational Hierarchy 94 example *and* use groups to represent projects like the Using Groups for Projects 95 example. This is supported and gives us all the advantages of both approaches.

Creating Groups
We create the groups to represent organizational hierarchy using the sync process as described in the Using Groups for Organizational Hierarchy 94 example. We create the groups to represent the projects manually as described in the Using Groups for Projects 95 example. PeerReview Complete remembers which groups were created manually and which were created using sync, so the sync process doesn't delete the manually created project groups.

Assigning Group Members
Instead of manually naming each user that is a member of each project group like we did in the Using Groups for Projects 95 example, we can indicate that all the members in an organizational hierarchy department are working on a project by making the organizational hierarchy group a child of the project group. This is much easier than assigning individual user members, but we lose some fine-grained control.

Associating groups with Reviews
All of the organizational hierarchy groups have "Allow associate with review 91" set to "No". The *Western*, *SciFi*, and *Medieval* project groups have "Allow associate with Review 91" to "Yes". We configure the singular group label 75 to "Project" and the plural group label 76 to "Projects".

When a user creates a review it will be associated with a project group. If the user's organizational hierarchy group is only a member of one project group then when they create a review they don't have to manually choose a group - the review is automatically associated with the project group. If a user's organizational hierarchy group is a member of more than one project group then when they create a review they are prompted to select one of those "Projects" to associate with the review.

Filtering the participant list
When a user creates a review they can choose to filter the participant list 88 by that group. This lets a user filter the list of participants by only the users working on the project associated with the review. This is useful at large companies to make the list of users more manageable.

© 2003-2011 SmartBear Software

Restricting Access based on group
Access to reviews can be restricted based on associated group. The restriction can be set for all reviews or just for a particular review. For example, if a review was associated with the *Western* group and group access restriction was enabled, then only users working on the *Western* project would be able to access it.

Reporting based on group membership

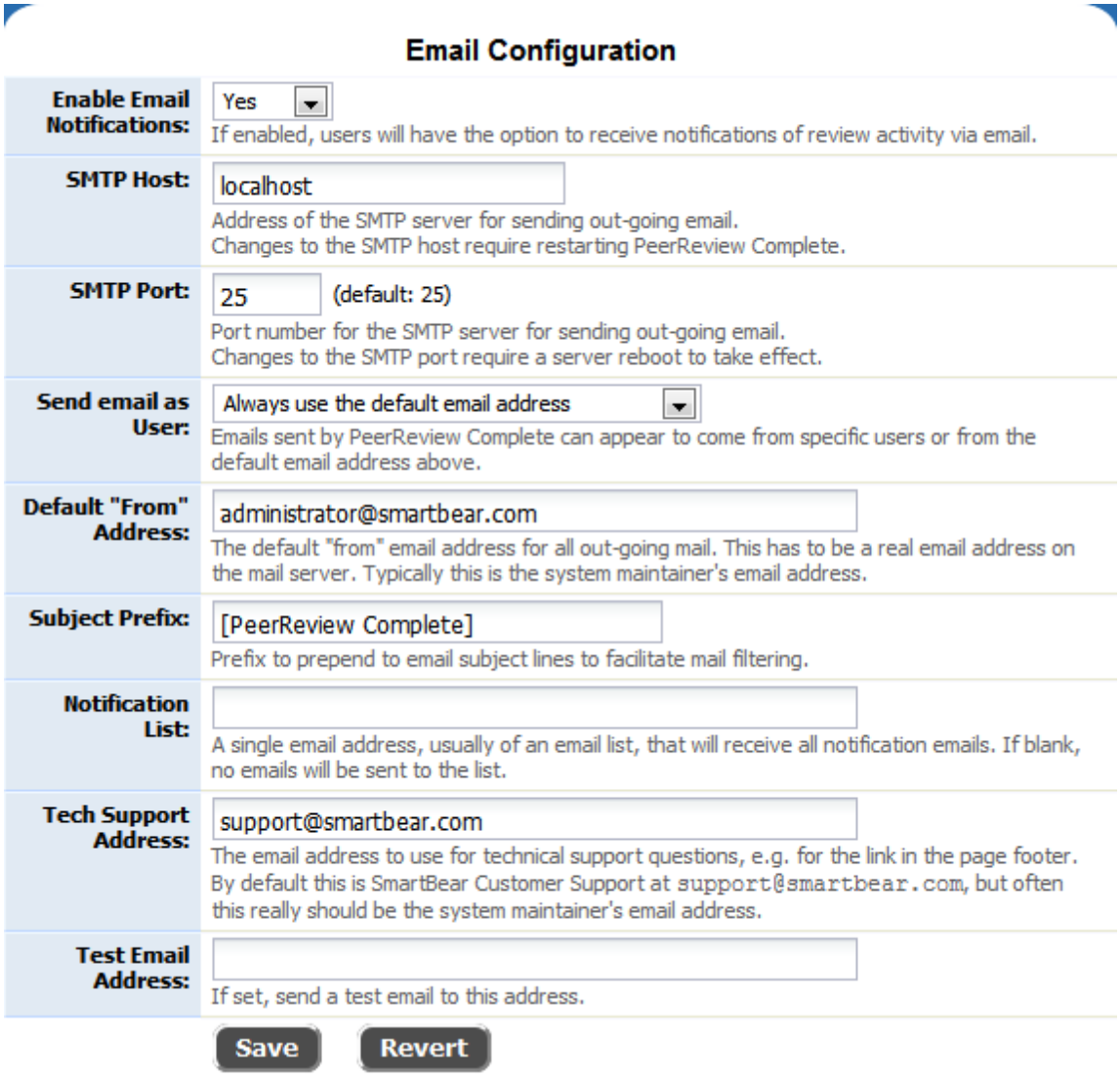
The reporting system can be used to create reports of e.g. all reviews where a member of a department was a participant, or all reviews created by a member of a department, etc...

Reporting based on associated group
You can also display, sort and filter reports by the associated group, so you know which project the review was for.

Finishing projects
When a project is finished, we mark it's group disabled. This means the group can no longer be associated with reviews, but it's still in the system for reporting purposes.

© 2003-2011 SmartBear Software

2.3.4 E-Mail Configuration


Email Configuration
Enable Email Notifications:
Yes
If enabled, users will have the option to receive notifications of review activity via email.
SMTP Host:
localhost
Address of the SMTP server for sending out-going email.
Changes to the SMTP host require restarting PeerReview Complete.
SMTP Port:
25
(default: 25)
Port number for the SMTP server for sending out-going email.
Changes to the SMTP port require a server reboot to take effect.
Send email as User:
Always use the default email address
Emails sent by PeerReview Complete can appear to come from specific users or from the default email address above.
Default "From" Address:
administrator@smartbear.com
The default "from" email address for all out-going mail. This has to be a real email address on the mail server. Typically this is the system maintainer's email address.
Subject Prefix:
[PeerReview Complete]
Prefix to prepend to email subject lines to facilitate mail filtering.
Notification List:
A single email address, usually of an email list, that will receive all notification emails. If blank, no emails will be sent to the list.
Tech Support Address:
support@smartbear.com
The email address to use for technical support questions, e.g. for the link in the page footer. By default this is SmartBear Customer Support at support@smartbear.com, but often this really should be the system maintainer's email address.
Test Email Address:
If set, send a test email to this address.
Save
Revert

Enable Email Notifications — Email notifications are now optional. If enabled, users will have the option to receive notifications of review activity via email.

SMTP Host — The domain name or IP address of the server to use to send SMTP messages. This machine must be configured to accept mail from the server on which PeerReview Complete is installed.

If your email server is Exchange you will need to get your Exchange administrator to enable anonymous SMTP.

If this setting isn't established properly, no emails will be sent.

© 2003-2011 SmartBear Software

SMTP Port

The port number to connect on when sending SMTP messages, typically 25. The port number might be different if a spam filter is set up in front of the primary server; in this case you probably want to bypass the spam machine, or else put a rule on the spam machine to allow PeerReview Complete to send messages.

If this setting isn't established properly, no emails will be sent.

Send Email as User

Determines which user will be listed in the "From Address" in emails sent by PeerReview Complete.

If you elect to use the "default address," emails will come from the default address given elsewhere in this configuration screen. Otherwise the system will attempt to send email from the user that caused the message to be sent. For example, if a user causes a "new review" notification to be sent, that user's email address will be used as the "from" address.

Some messages cannot be associated with a particular user and will still be sent from the default address.

Default "From" Address

The email address to use in the "From Address" field in notification emails. Typically this is the system maintainer's email address.

Subject Prefix

This text will be pre-pended to the subject line of any email sent by PeerReview Complete. This assists end users with mail filtering and helps to train or configure spam-filters to identify PeerReview Complete notifications as non-spam.

Notification List

This email address will receive a copy of *every* notification sent out by the system. It is often left blank. This address can also be a mailing list.

Tech Support Address

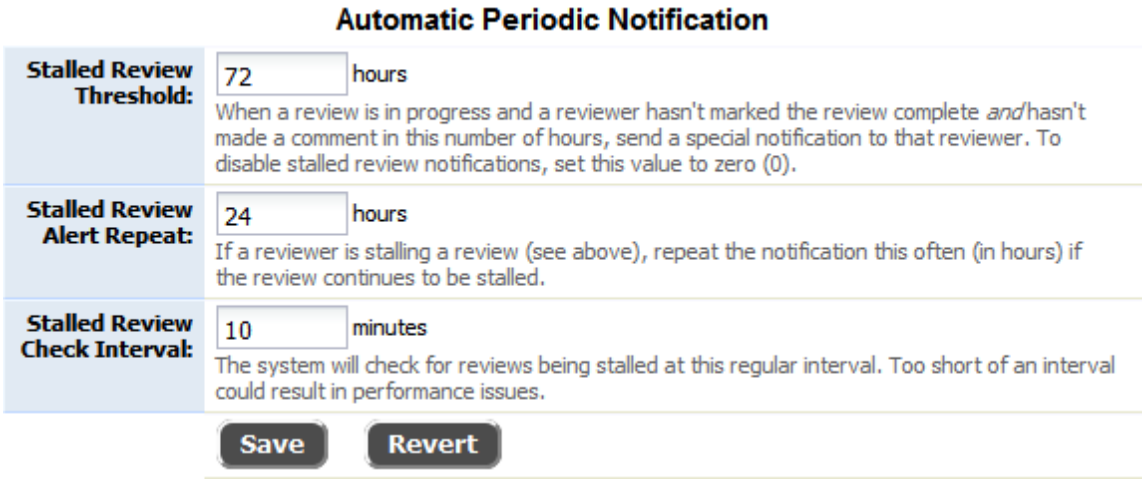
The email address to use for technical support questions. For example, this is used in the `[Support]` link in the web page footer.

By default this is the email address for PeerReview Complete Technical Support, but often the system maintainer wants to get these emails first for internal resolution.

Test E-Mail Address

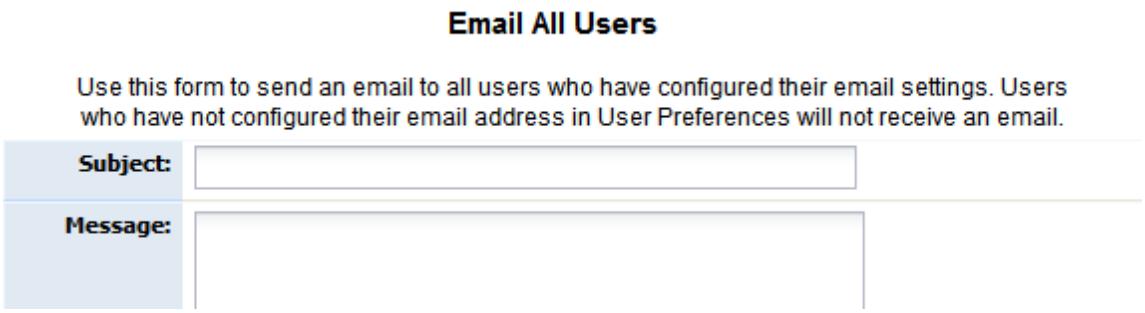
Usually left blank; if an address is supplied here a test message will be sent to this address when the form is submitted. This is used to test the email system. The email address is not saved.

© 2003-2011 SmartBear Software


Automatic Periodic Notification
Stalled Review Threshold:
72
hours
When a review is in progress and a reviewer hasn't marked the review complete and hasn't made a comment in this number of hours, send a special notification to that reviewer. To disable stalled review notifications, set this value to zero (0).
Stalled Review Alert Repeat:
24
hours
If a reviewer is stalling a review (see above), repeat the notification this often (in hours) if the review continues to be stalled.
Stalled Review Check Interval:
10
minutes
The system will check for reviews being stalled at this regular interval. Too short of an interval could result in performance issues.
Save
Revert

Stalled Review Threshold When a review is in progress and a reviewer hasn't marked the review complete and hasn't made a comment in this number of hours, send a special notification to that reviewer. To disable stalled review notifications, set this value to zero (0).

Stalled Review Alert Repeat If a reviewer is stalling a review (see above), repeat the notification this often (in hour if the review continues to be stalled.


Email All Users
Use this form to send an email to all users who have configured their email settings. Users who have not configured their email address in User Preferences will not receive an email.
Subject:
Message:

Bulk-Email Facility

At the end of the email configuration dialog is a form that allows the administrator to send an email to all users 84 in the system. This is easier than maintaining a separate email mailing list for users.

A typical use is to broadcast scheduled maintenance of the system, especially before and after system upgrade 37.

2.3.5 Version Control

PeerReview Complete integrates closely with your version control server to select which files to review. The administrator can configure the PeerReview Complete server to be able to connect directly with version control. This allows users to select checked-in changelists to review from the web UI without

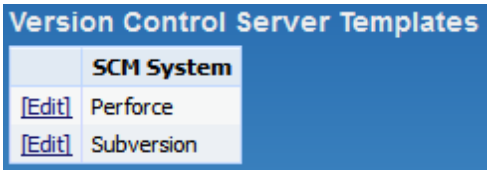
© 2003-2011 SmartBear Software

having to install and configuration client-side tools. Users can also configure their connection to the various version control servers using the client tools 182 so they can review changes before committing them.

If you use more than one version control server PeerReview Complete recognizes this and displays the files from the different servers separately.

Configuring Version Control Server Templates

Each version control system that PeerReview Complete supports for server-side integration has a configurable "template". The configuration in this template is copied in to a version control server entry when you click ***Create*** in the **Configure a new version control server** form, or when a version control server entry is created automatically because a client uploaded files from a server that doesn't map 103 to any of the existing version control server entries.

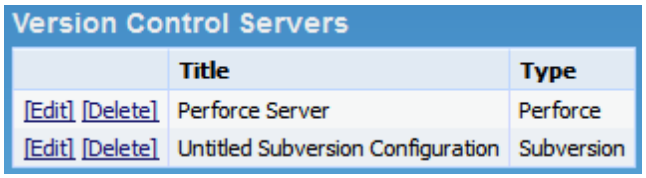


Each version control system that supports server-side version control integration includes different options in the Version Control Server Template:

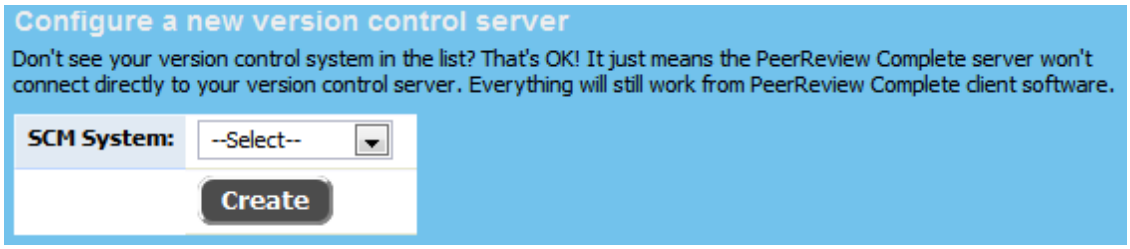
- Perforce 367
- Subversion 397

Configuring Servers

Next is the list of currently configured servers, or a message if you haven't created any yet:



Entries in this table are created automatically when one of the client tools uploads files from a server that doesn't map 103 to any of the servers we currently know about. You can edit or delete entries in this list. A form to create a new entry always appears at the bottom of the screen:


Configure a new version control server
Don't see your version control system in the list? That's OK! It just means the PeerReview Complete server won't connect directly to your version control server. Everything will still work from PeerReview Complete client software.
SCM System:
--Select--
Create

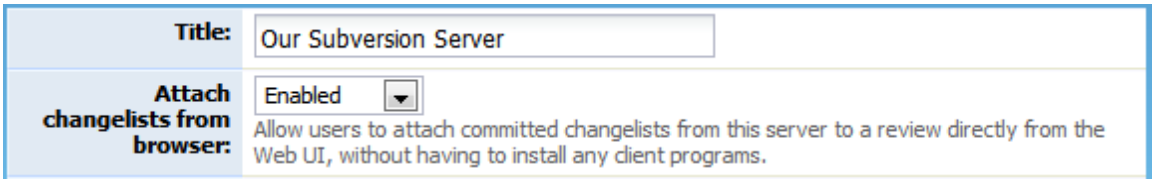
© 2003-2011 SmartBear Software

Editing Server Configuration

The exact fields you need to configure the connection to a version control server depends on the version control system:

- Perforce 367
- Subversion 397

All the version control server entries have at a minimum these two fields:


Title:
Our Subversion Server
Attach changelists from browser:
Enabled
Allow users to attach committed changelists from this server to a review directly from the Web UI, without having to install any client programs.

Title The title is used to indicate this version control server to other users of PeerReview Complete, so it should be something that everyone will understand, even if they're going through proxies, VPNs, or other such things.

Attach changelists from browser If enabled, this feature lets users select committed changelists to review directly from the web browser, without having to install any client programs.

All of the version control systems also require that you specify the path to the command-line client executable for that system. This is a good example of something you might want to specify in the version control server template 102, since it's probably going to be the same for all the version control server entries you configure.

If you want to enable the Attach changelists from browser 103 feature, then the server will also require some sort of authentication. This is the version control server account that PeerReview Complete should use when querying that version control server. Note that PeerReview Complete will use this account no matter who is logged in!

You can click ***Test Connection*** at the bottom of the form to make sure that PeerReview Complete can successfully communicate with the server.

Client Configuration Mapping

An unfortunate thing happens when client tools are separately configured for version control access: they can have different names for the same physical version control server. It's hard or impossible for the PeerReview Complete server to automatically figure out whether these two names represent the same server or two different version control servers.

The administrator can fix this using the **Client Configuration Mapping** section of the server configuration form. You can configure java-style regular expressions to match against each of the fields that our client tools upload. If the version control server entry was created automatically then these fields are already filled out and you probably don't need to change them. The fields available are different for each version control system:

- Perforce 368
- Subversion 398

© 2003-2011 SmartBear Software

When you submit this form, the list of configurations updates to indicate which configurations match and which don't match the current patterns. This makes it easy to iterate until you've covered exactly the right cases.

2.3.6 Custom Fields

PeerReview Complete supports custom fields for reviews and defects. A special type of review custom field is also supported: participant custom fields. Participant custom fields apply to a review, but allow each user to specify their own value for the field. Custom fields may be any of four types: Single-line String, Multi-line String, Drop-down List, and Multi-Selection. Although these fields show up in different places in the user interface, the administrative interface for configuring these fields is identical. It shows up in the `Review Custom Fields`, `Participant Custom Fields`, and `Defect Custom Fields` subsections of the administrative interface.

Consequences of Changing Custom Fields

The conscientious administrator will realize that it could be disastrous to change custom field settings while a review is taking place. If someone is filling out custom fields and simultaneously a new one is added, what happens? If a review is going on and a new field is added or a field is removed, what happens? If you view an old review with defects and you have since added, removed, or changed defect custom fields, what happens to the defect data?

PeerReview Complete is smart about handling dynamically changing custom fields even while reviews are in progress or already complete. If fields are added after the fact, reviews and defects will show a dash or "N/A" to indicate that this value wasn't given. Any new creations or edits of existing creations will force the user to fill in values for the new fields. If fields are deleted, existing values are simply ignored. If there are changes to field titles, descriptions, or other non-value items, existing values are unaffected. If the values in drop-down lists change such that a value given previously no longer exists, the previous value will still hold for that field, although if the user then edits the field he will be forced to pick a new valid value.

Fields that are deleted in the administrative interface are removed from the user's interface immediately. The original data is still preserved in the database for reporting but is no longer available to the user in the various client interfaces.

Custom Fields and Reports

The built-in reports 176 do not reference custom fields. There are, however, consequences of custom fields on external reporting systems such as Excel, Access, Crystal Reports, and Business Objects.

The custom field system inside the database is fairly involved and not intuitive. Therefore for your convenience PeerReview Complete maintains several database VIEW objects that cause custom field data to appear as if it was a standard table in the database. The two VIEW objects are `reviewcustom` and `defectcustom`. In each case one column refers to the unique ID of the associated object, then PeerReview Complete generates columns for each of the other custom fields and fills in the values for each, using `NULL` if the value isn't specified.

© 2003-2011 SmartBear Software

Warning: It is not uncommon to see `NULL` values in custom field views. If a review is in a workflow [112] that doesn't support certain fields, they will be `NULL`. If a review was created before a certain custom field existed, the value will be `NULL`.

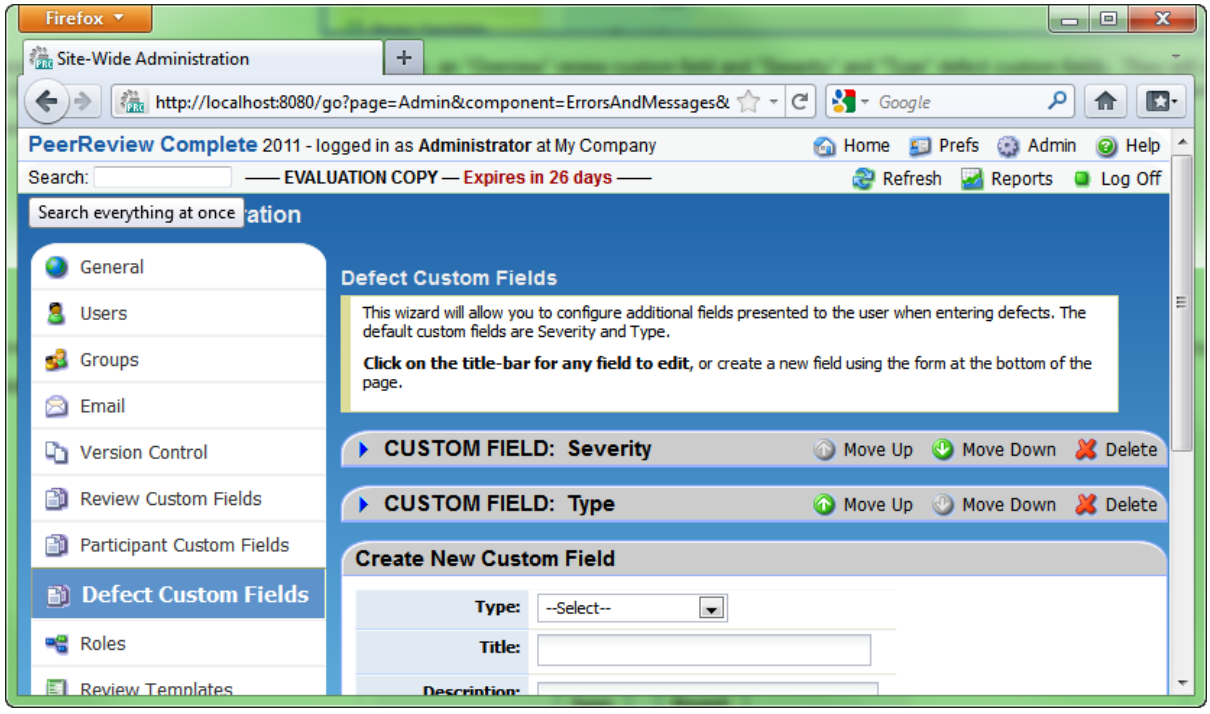
Warning: If you change custom fields PeerReview Complete will immediately change the VIEW definitions in the database. This can break existing queries and reports unless those queries have been designed to be immune to such changes. For example, you should refer to columns by name and not by their order.

Custom Fields and Searching

The search engine [174] is aware of all custom fields and will scan data in custom fields you would expect.

Custom Field Configuration

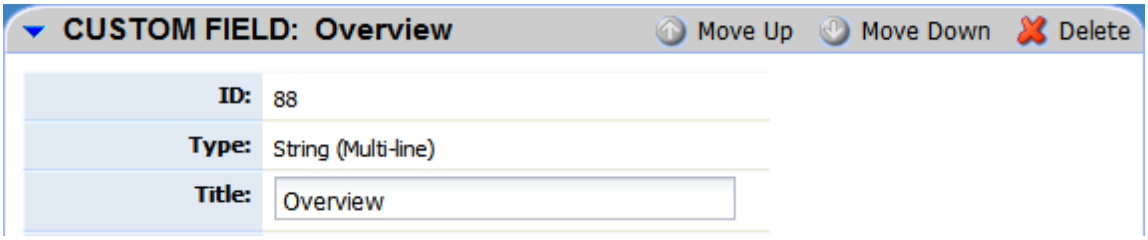
When you view either the `Review Custom Fields`, `Participant Custom Fields`, or `Defect Custom Fields` subsection, a list of already customized fields will appear at the top of the list:


Firefox
Site-Wide Administration
http://localhost:8080/go?page=Admin&component=ErrorsAndMessages&
PeerReview Complete 2011 - logged in as Administrator at My Company
Home Prefs Admin Help
Search:
EVALUATION COPY — Expires in 26 days
Refresh Reports Log Off
Search everything at once
General
Users
Groups
Email
Version Control
Review Custom Fields
Participant Custom Fields
Defect Custom Fields
Roles
Review Templates
Defect Custom Fields
This wizard will allow you to configure additional fields presented to the user when entering defects. The default custom fields are Severity and Type.
Click on the title-bar for any field to edit, or create a new field using the form at the bottom of the page.
CUSTOM FIELD: Severity
Move Up Move Down Delete
CUSTOM FIELD: Type
Move Up Move Down Delete
Create New Custom Field
Type:
--Select--
Title:

PeerReview Complete will come with default custom fields, an "Overview" review custom field and "Severity" and "Type" defect custom fields. They will appear collapsed as exemplified by the Severity custom field. Click on the blue triangle to the left of the custom field title, and the section will expand so that you may view/edit the settings. Expanded sections will look like the Type custom field.

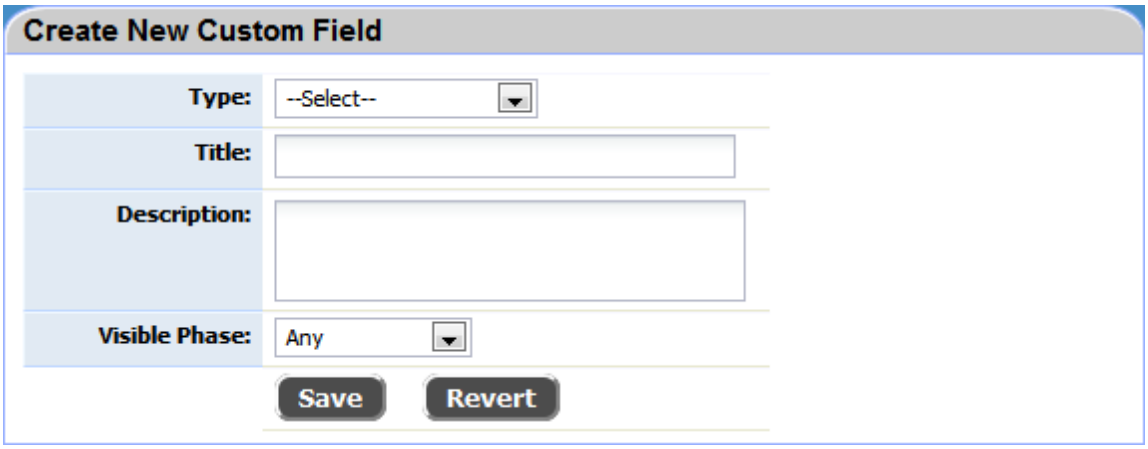
© 2003-2011 SmartBear Software

Each existing custom field comes in a box with a toolbar of commands to the right of the title:



The `Move Up / Move Down` commands let you change the order of the fields as it will appear to users. `Delete` removes the custom field from the system forever, so use with caution.

At the bottom of the custom fields list, there will be a customizable form where you will be able to create a new custom field:



A custom field's data type cannot be changed once it is created; everything else about the field can be modified at any time. Types are one of:

- **Single-Line Text**. A single line of text, scrolling horizontally if needed for length. Maximum length is 255 characters. Various validation options.
- **Multi-Line Text**. Multiple lines of text, scrolling horizontally and vertically as needed. Maximum length is database-dependant, but always at least 8192 characters, and typically more like many millions of characters.
- **Drop-Down List**. A drop-down list of pre-defined choices. Exactly one selection is required.
- **Multi-Select List.** A drop-down list of pre-defined choices. User may select zero, one, or multiple values.

The Title of a field is displayed to the user as a prompt, much like "Type" and "Title" are shown in the screenshot above. The Description of a field is shown as help-text below the field, prompting the user as to the intent of the field.

All textual fields allow for a minimum and maximum length. Minimums can be zero if it's OK for the user to leave the field blank.

© 2003-2011 SmartBear Software

All textual fields also allow for an optional Java-style regular expression validator. If given, the content of the field must match the regular expression exactly. It is important to note that if you use a Validator, that field will be automatically required. If you do not want the field to be required, you must use a Validator that will allow the field to accept empty values AND set the Minimum Length to 0.

The Drop-Down List type requires a list of options. These are in plain text. To change the order of items, just copy/paste the items in the list. If you change just capitalization, the item is altered in-place; if you change an item more than just in capitalization the system considers the previous item to be deleted and the current item to be added.

> **Warning:** If you think you're renaming a drop-down item, you're not! The system considers "renaming" as deleting the old item and creating a new item. Rearranging items without renaming works as expected, as does changing just capitalization in an item.
>
> This confusing situation will be remedied in a future version.

Why Aren't My Custom Fields Showing Up in My Reviews?
Once you've created a new custom field, you will not see them automatically as options in your reviews. Custom fields will be automatically be configured as "Hidden" under all templates. To show custom fields, you must configure them to "Show" under the Workflow Configuration 114.

You can also define custom fields to be visible only during certain phases of the review. This can be useful for storing fields only used by scripts or triggers. This is changed via the "Visible Phase" drop-down.

Notes About Participant Custom Fields
Participant custom fields have some unique behaviors. First, they cannot be visible only in the planning phase, as this would not give the participants an opportunity to specify their values for the fields. Second, if a required participant custom field is visible only within the Inspection phase, users may not "Finish" the inspection phase without specifying their values for the fields. Finally, participant custom fields may be edited while the review is in the complete phase without reopening the review. This is to support post-review survey questions.

For examples of reporting on Participant Custom Fields, see:

- Example SQL: Participant Custom Fields 435
- Example XPath and XSL 441

2.3.7 Role Configuration

A "Role" in PeerReview Complete indicates the way in which a person participates in a review.

Roles
PeerReview Complete supports four "roles" for review participants. The following roles are configured by default but administrators can configure completely different templates.

© 2003-2011 SmartBear Software

Author
Authors are the people responsible for the changes or documents under review. Usually they are responding to comments and questions made by reviewers and observers. In the "Fix Defects" phase the review will be "in their court" and on theirAction Items 139 list.

Reviewer
Reviewers are responsible for inspection, creating defects, etc.. With multiple reviewers, they will hopefully come to a consensus about each questionable item, but this is not required by the system. Reviewers typically create defects 170, but again this is not required.

Observer
Observers are involved and make comments but they are not "vital" to the review. If all other participants mark the review "complete," the review goes to the next phase regardless of observer state. Observer roles are usually used when you want to bring in someone who has special expertise on an issue.

Moderator (optional, not enabled by default)
The Moderator maintains the pace and tenor of the review. This is an optional role that doesn't exist in the default installation. This is used for more formal review workflows where one person leads and controls the review..

Different styles of review require different roles with different terminology and rules for what each rule is allowed to see and do in a review. Formal reviews might have four roles with strict rules, informal reviews might have just an author and reviewer, and a "self-check" review might just require an author with optional external reviewer.

Users interact with roles when they arecreating a new review 142 or editing the list of participants 150 in a review.

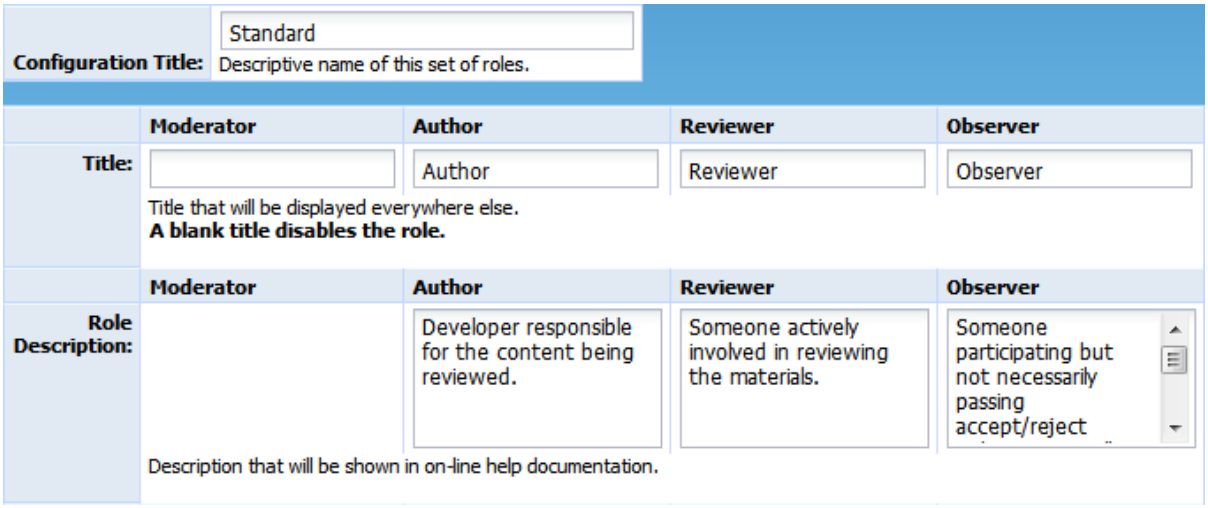
The Role Configuration screen lets you set up any number of sets of roles. Each set is given a name and corresponds to some concept of a review.

Any number of role configurations can be specified. The names here are never shown to an end user; they are used only when selecting a role configuration as part of areview workflow 112.

In each role configuration you can have between 1 and 4 roles.

© 2003-2011 SmartBear Software

A variety of options are available when editing an existing role configuration or creating a new one:



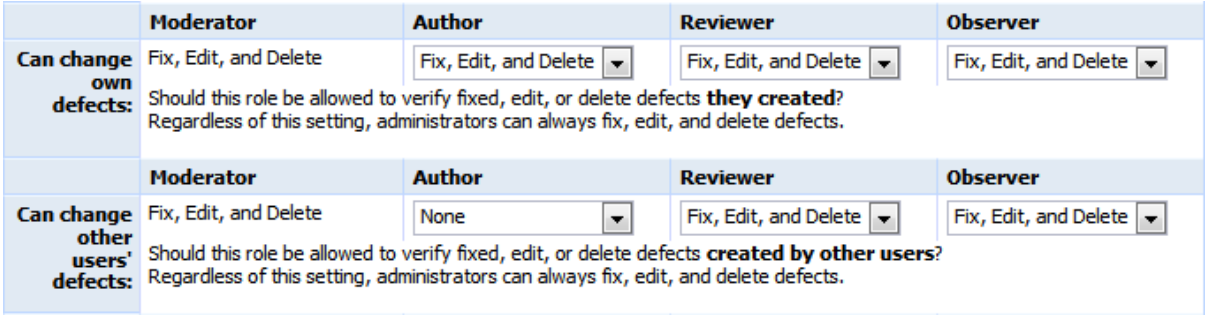
Configuration Title — The title of this configuration. This is strictly an administrative thing -- users of the system will never see this title. They see review workflows [112] instead.

Title — The name of the role as it will appear in drop-down lists and on-line help text. The user will use this title as the primary mechanism for referring to the role.

Use a blank title to indicate that this role should be completely disabled.

The "Author" role *must* always be in the second column, although it can be renamed. It *must* always exist. The concept of the "author" is a special one in the system because the author always has special responsibilities. For example, when a review is in "Rework" phase the author's Action Items [139] list will say "Rework defects found" whereas other participants will see "(No Action Required) Waiting for author to rework defects."

Description — A description of the role that will be prominently displayed to the user in all client user interfaces where roles are chosen, most prominently in the Participants [142] section of the Create Review wizard.


Moderator
Author
Reviewer
Observer
Can change own defects:
Fix, Edit, and Delete
Fix, Edit, and Delete
Fix, Edit, and Delete
Fix, Edit, and Delete
Should this role be allowed to verify fixed, edit, or delete defects they created?
Regardless of this setting, administrators can always fix, edit, and delete defects.
Moderator
Author
Reviewer
Observer
Can change other users' defects:
Fix, Edit, and Delete
None
Fix, Edit, and Delete
Fix, Edit, and Delete
Should this role be allowed to verify fixed, edit, or delete defects created by other users?
Regardless of this setting, administrators can always fix, edit, and delete defects.

© 2003-2011 SmartBear Software

Can change own defects

If "None", this role will be allowed to create defects, but will not be allowed to mark their own defects fixed, edit their own defects, or delete their own defects.

If set to "Fix and Edit", this role will be able to create defects, mark their own defects fixed, and edit their own defects, but will not be allowed to delete their own defects.

If set to "Fix, Edit, and Delete", this role will be able to create defects, mark their own defects fixed, edit their own defects, and delete their own defects

Administrators can always change or delete any defect, regardless of this setting.

In all but the most informal of reviews, authors are typically not allowed to mark defects fixed, but all other roles are allowed.

This setting also applies to the ability to externalize 171 a defect.

Can change other users' defects

If "None", this role will not be allowed to mark other users' defects fixed, edit other users' defects, or delete other users' defects.

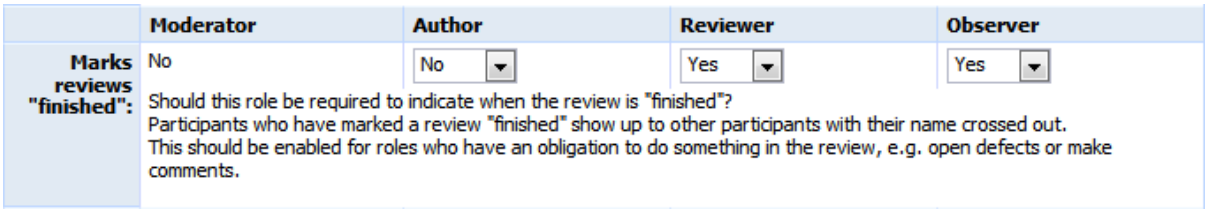
If set to "Fix and Edit", this role will be able to mark other users' defects fixed and edit other users' defects, but will not be allowed to delete other users' defects.

If set to "Fix, Edit, and Delete", this role will be to mark other users' defects fixed, edit other users' defects, and delete other users' defects

Administrators can always change or delete any defect, regardless of this setting.

To enable this setting, the Restrict Access to Fix Defect 77 setting must be set to "No".

This setting also applies to the ability to externalize 171 a defect.


Moderator
Author
Reviewer
Observer
Marks reviews "finished":
No
No
Yes
Yes
Should this role be required to indicate when the review is "finished"?
Participants who have marked a review "finished" show up to other participants with their name crossed out.
This should be enabled for roles who have an obligation to do something in the review, e.g. open defects or make comments.

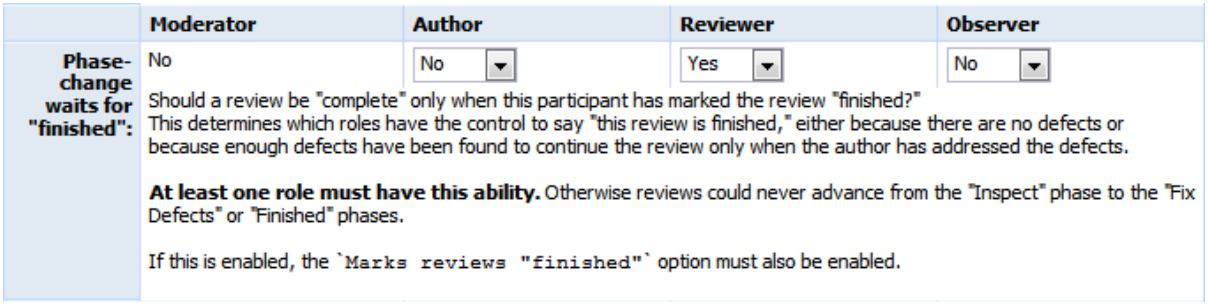
Marks reviews "finished"

If true, users with this role will be presented with a button in the Moving On screen 156 that allows the user to say to the group: "I am finished looking at this review." The user will be brought back into the review if another user makes a non-trivial comment.

If false, the user is told "You must wait for other users."

© 2003-2011 SmartBear Software

Typically everyone except the author has this feature enabled. The author is usually not in control of when the review finished so his "I'm finished" decision is not interesting.

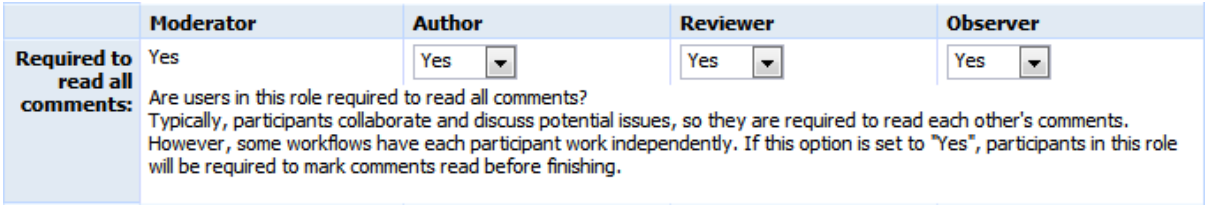

Moderator
Author
Reviewer
Observer
Phase-change waits for "finished":
No
No
Yes
No
Should a review be "complete" only when this participant has marked the review "finished?"
This determines which roles have the control to say "this review is finished," either because there are no defects or because enough defects have been found to continue the review only when the author has addressed the defects.
At least one role must have this ability. Otherwise reviews could never advance from the "Inspect" phase to the "Fix Defects" or "Finished" phases.
If this is enabled, the `Marks reviews "finished"` option must also be enabled.

Phase-change waits for "finished"

If true, the review does not change phases until this user says he is "finished" with the review in the Moving On screen 156. A phase-change might be inspection -> rework or inspection -> complete.

For this to be true, "Marks reviews finished" must also be true.

Typically this will be enabled for reviewers but disabled for observers where the user has been invited but we don't want the review held up.


Moderator
Author
Reviewer
Observer
Required to read all comments:
Yes
Yes
Yes
Yes
Are users in this role required to read all comments?
Typically, participants collaborate and discuss potential issues, so they are required to read each other's comments. However, some workflows have each participant work independently. If this option is set to "Yes", participants in this role will be required to mark comments read before finishing.

Required to read all comments

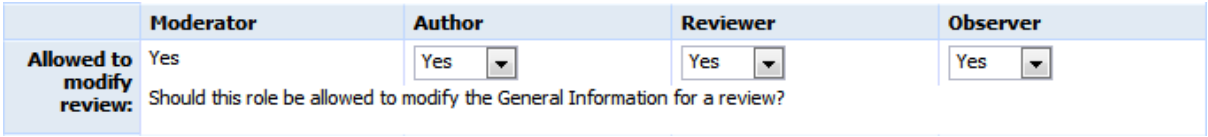
If yes, participants in this role will be required to mark comments read before finishing the review.

Typically participants collaborate and discuss potential issues, so they are required to read each other's comments. However, some workflows have each participant work independently.

Setting this configuration to "no" is usually applied to observer-type roles because they are frequently called in for something specific and don't need to "hang around" and see what other users have to say. This is also helpful in workflows where the reviewers independently submit comments to the author, who is then in charge of all changes without having to communicate back to reviewers.

Note that this setting doesn't prevent anyone from coming back to the review and continuing to actively participate at any time.

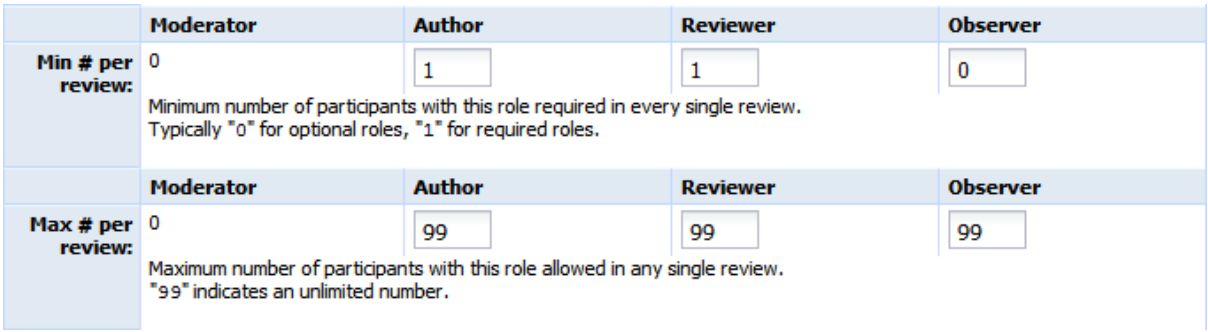
© 2003-2011 SmartBear Software



Allowed to modify review

Should this role be allowed to modify the review title and custom fields?

Typically there is no harm in allowing this, but you might want to lock down exactly who has the last word on the review custom fields and title.



Min # per review

The minimum number of users with this role required in the review. Can be zero to indicate "This role completely optional." Most often used with reviewers or moderators to require at least one, or with a "very careful" review where multiple reviewers are required.

Max # per review

The maximum number of users with this role required in the review. Use 99 to indicate "no maximum." Usually there is no reason to limit the number of users who can participate as a role in a review, except for highly formal reviews where a strict process is required.

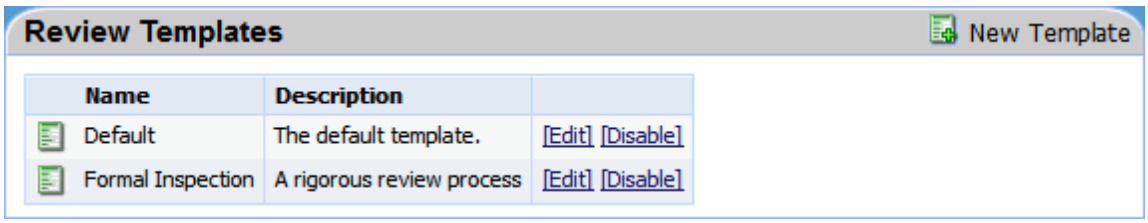
2.3.8 Workflow Configuration

A "Review Template," or a "Review Workflow," is a particular set of roles, custom fields, and other options that determine the behavior and rules of a review.

Small installations might have only one template. Some installations will want a template for "quick" reviews (self-checks during active development of new features), another for "standard" (one author, one reviewer, suitable for most reviews), and another for "very strict" (reviews of a stable branch, core module, stable files or particularly difficult piece of code). Still others might require different templates in different development groups all using a single server.

© 2003-2011 SmartBear Software

The Review Templates section in the Admin screen allows an administrator to configure these templates.



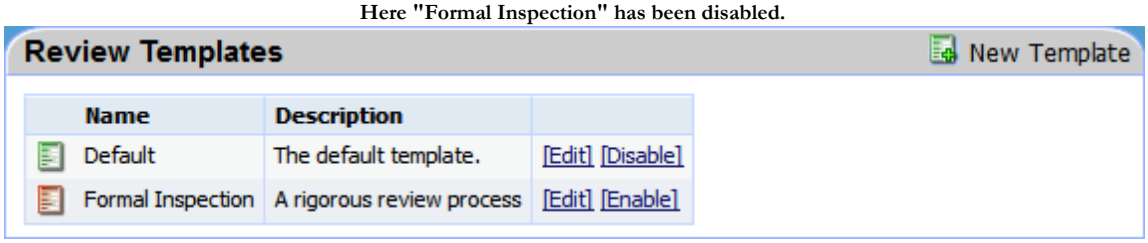
Template Management

The default configuration is to have only a single active template, and this is adequate for many smaller installations and trials. When there is exactly one template, the user is not prompted to select a template when he creates a review[141]; the one template is automatically selected. If additional templates are added later, the template that was the singleton is still applied to those existing and completed reviews.

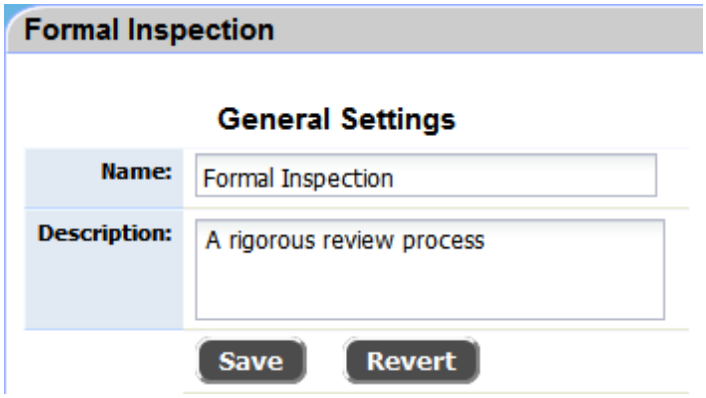
You cannot have fewer than one active template. The user interface will not allow you to disable the last template in your system; users need *something* to control the rules of the review.

You also cannot delete templates. Old templates are needed for existing and completed reviews that depended on that template. You can, however, mark templates as "disabled," in which case they appear only in the administrative interface but never when creating a new review. Existing reviews with that template continue to their completion so as to not disrupt their progress:

Here "Formal Inspection" has been disabled.

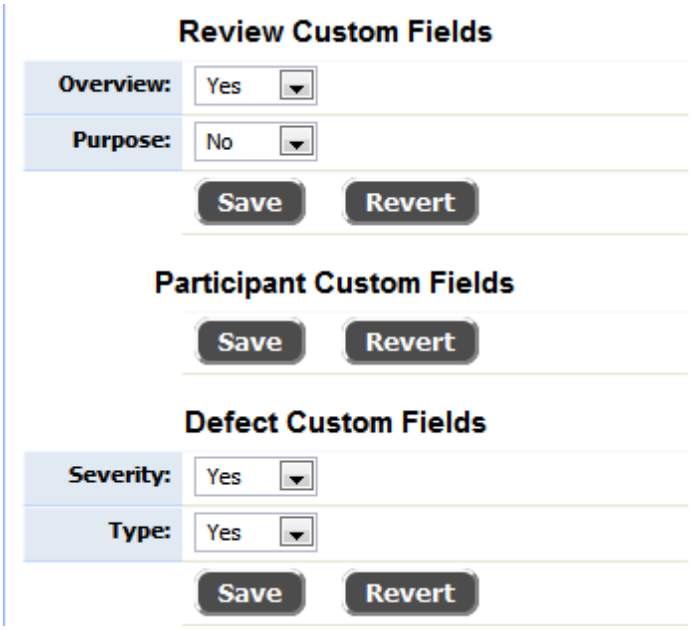


Editing Templates

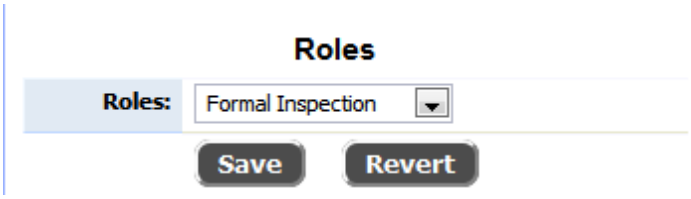


Every template has a Name and Description. Both of these are shown to the user for reference.

© 2003-2011 SmartBear Software



Each template can include an arbitrary set of custom review and defect fields 104. The master list of all possible custom fields is set up separately; here you can configure which of those fields to display to the user during this template. This makes it possible to have templates with various field requirements, but where the same field *can* be reused across templates, the database and user interface will make that plain.



Each template also specifies a single role configuration 107 to use for that review.

Templates in Flux

What happens when a review changes its template, or when a template definition changes when a review is already in progress or complete?

A review can only change template during the Planning phase. This means once the review is under way, the template is fixed. This makes sense -- it would be hard to change e.g. the role configuration 107 when participants are already invited and performing their respective tasks! During the planning phase the wizard interface prevents the user from making incorrect moves.

If template definitions change while a review is in progress, the system gracefully adjusts in a similar manner to changing custom fields 104. Existing data is saved; if new fields are present they will be required when a new review or defect is created but are not required if a review or defect is already created. If a custom field is removed from the template it will disappear from the review interface immediately. But the data is not lost, and if the custom field is re-enabled in that template it will re-appear with the content it had previously.

© 2003-2011 SmartBear Software

When a custom field is added and there is exactly one active template, that template will be automatically updated to include that field. If there is more than one active template, none are updated automatically and the default is to "not show" the field, so the administrator will need to visit all applicable templates and add this field to the list.

In general the system handles changes to templates quite gracefully, even if reviews are already in progress. Still, in the interest of not surprising users, it's usually best to change template rules on off-hours, and to send an explanatory email 101 so users are aware of what's going on.

2.3.9 Triggers

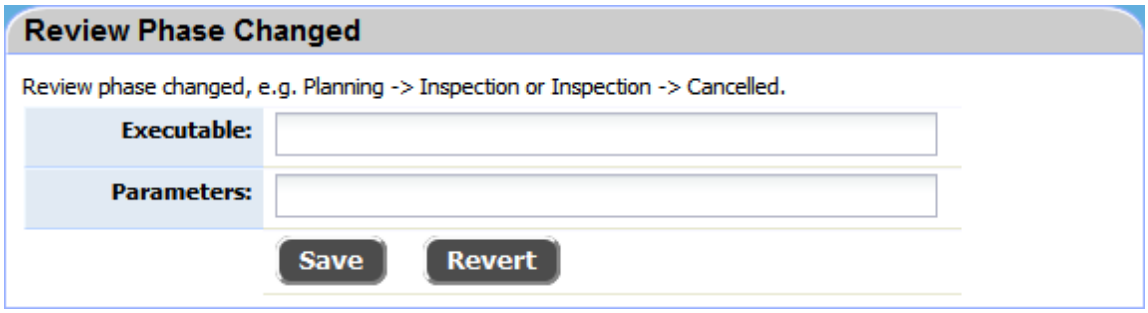
PeerReview Complete can execute an external script or application when certain events occur. This is useful when integrating PeerReview Complete with other systems such as issue-tracking, ALM, or builds.

Possible uses of triggers include:

- Mirror defects created in PeerReview Complete into an external issue-tracking system.
- Set a flag in a system when a review completes.
- Intercept all notifications sent from PeerReview Complete to log them or to send the notification in a way other than standard email or RSS.
- Send notifications to a mailing list when certain events occur.

The Triggers section of the Administration page lets you configure the various triggers:

- **Review Created** - Review created. No files or participants yet.
- **Review Phase Changed** - Review phase changed, e.g. Planning -> Inspection or Inspection -> Cancelled.
- **Added Files** - Files and/or changelists attached to a review.
- **Notifications Sent** - A notification (email, RSS, etc.) was sent to someone.
- **Defect Activity** - A defect was added, modified, or marked open/fixed. Use the defect unique ID to determine the difference.
- **User Created** - A new user was created. This runs whether or not the creation was automated or manual.



© 2003-2011 SmartBear Software

The `Executable` field specifies the application to run. If this is a script, use the script executable here (e.g. Perl, PHP, Python). This must be an absolute path to the application; the form will complain if the file cannot be found or executed.

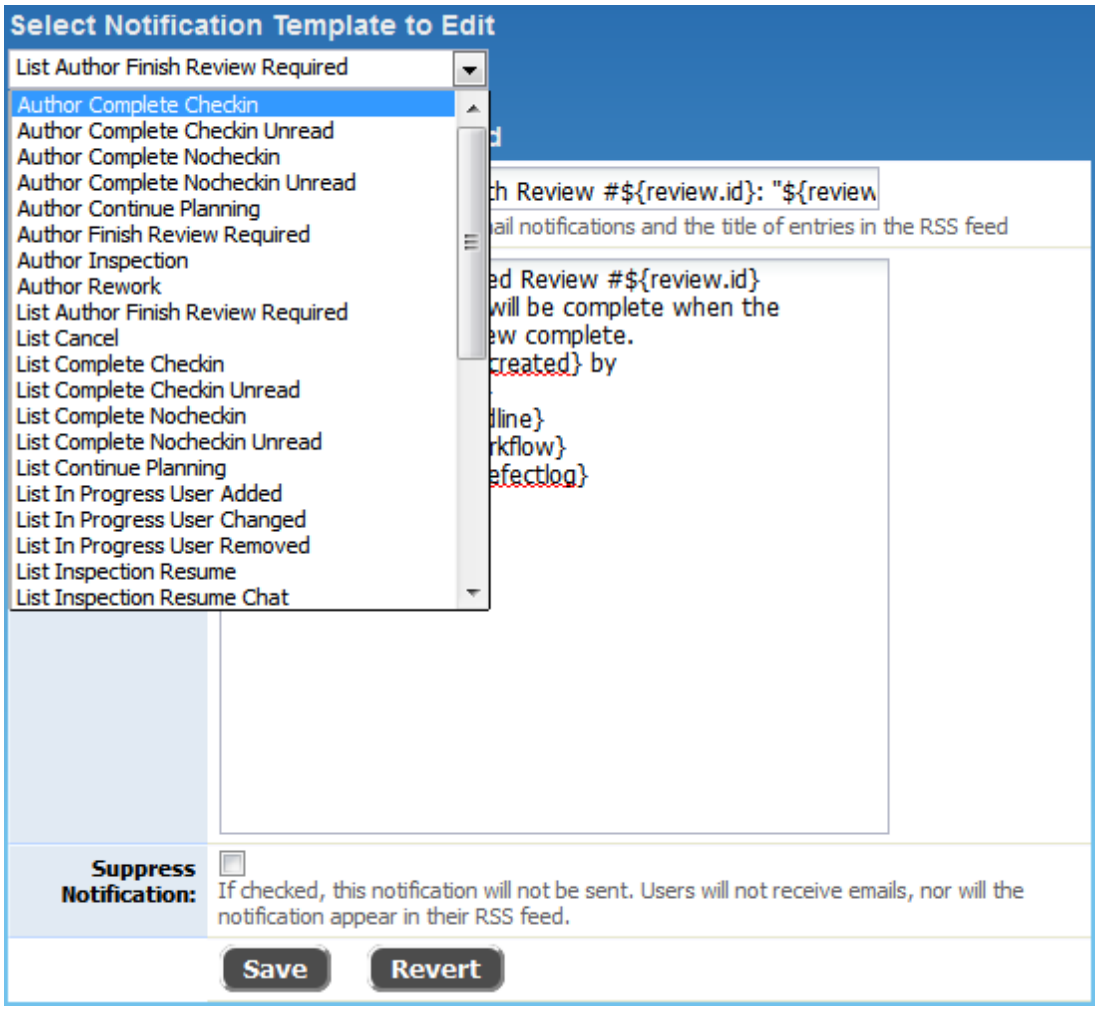
The `Parameters` field is a list of arguments to supply to the executable. If this is a script, the absolute path to the script itself is typically the first argument.

There are many special arguments you can supply that will be replaced by PeerReview Complete when the application is executed. For example,`${review.title}` will be replaced by the title of the review associated with the given triggering event. See theVariable Substitution 119 chapter for details.

2.3.10 Notification Templates

You can customize the content within the notification emails 134 sent by PeerReview Complete.

To do so, select the notification email you want to customize by using the drop down menu:


Select Notification Template to Edit
List Author Finish Review Required
Author Complete Checkin
Author Complete Checkin Unread
Author Complete Nocheckin
Author Complete Nocheckin Unread
Author Continue Planning
Author Finish Review Required
Author Inspection
Author Rework
List Author Finish Review Required
List Cancel
List Complete Checkin
List Complete Checkin Unread
List Complete Nocheckin
List Complete Nocheckin Unread
List Continue Planning
List In Progress User Added
List In Progress User Changed
List In Progress User Removed
List Inspection Resume
List Inspection Resume Chat
Suppress Notification:
If checked, this notification will not be sent. Users will not receive emails, nor will the notification appear in their RSS feed.
Save
Revert

A list of all possible notification emails will appear in the menu. Generally, recipients of the notification email will be denoted by the following key words:

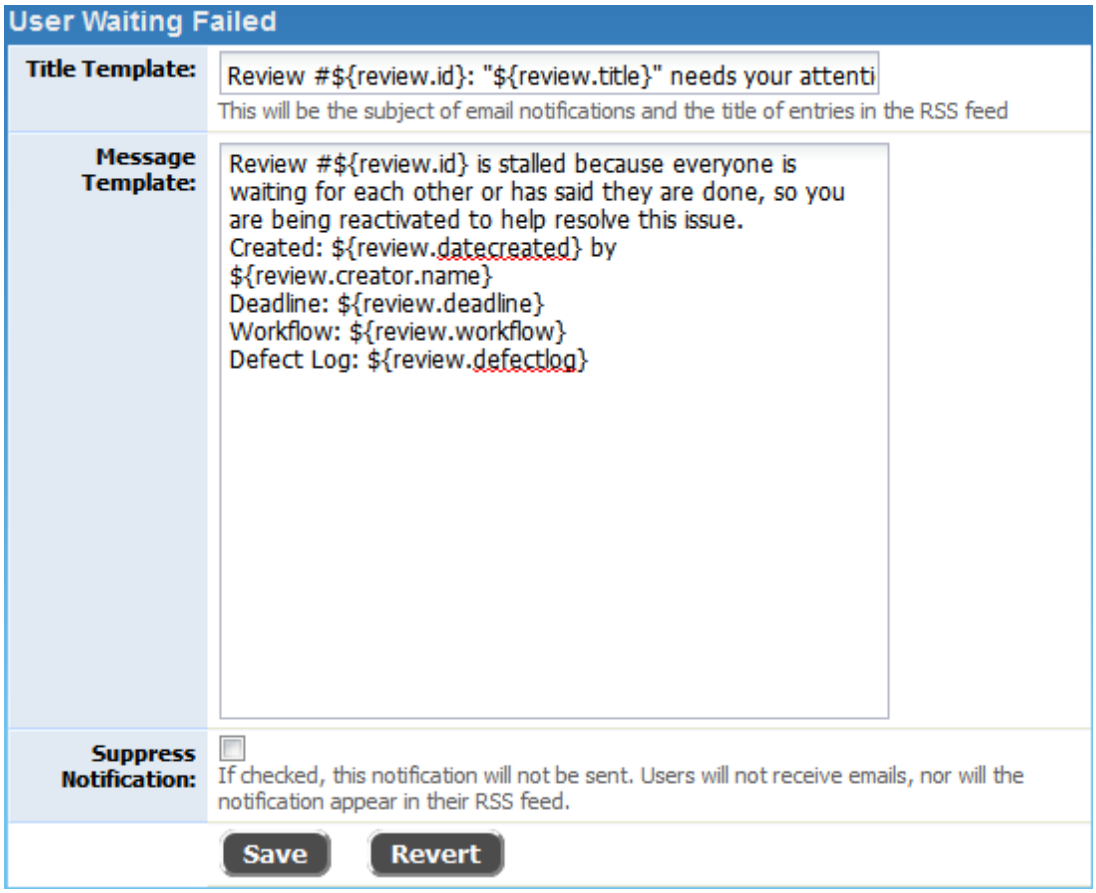
Author Author of the review

© 2003-2011 SmartBear Software

User	Review Participants
List	The notification list 100
Reviewer	Reviewers
Poke	Single user who is poked 150

Notification templates also describe when the particular email is sent out with their title. These are self-explanatory for the most part.

Once a template has been selected, the form below will change accordingly, and you may edit the template as desired:


User Waiting Failed
Title Template:
Review #${review.id}: "${review.title}" needs your attenti
This will be the subject of email notifications and the title of entries in the RSS feed
Message Template:
Review #${review.id} is stalled because everyone is waiting for each other or has said they are done, so you are being reactivated to help resolve this issue.
Created: ${review.datecreated} by ${review.creator.name}
Deadline: ${review.deadline}
Workflow: ${review.workflow}
Defect Log: ${review.defectlog}
Suppress Notification:
If checked, this notification will not be sent. Users will not receive emails, nor will the notification appear in their RSS feed.
Save
Revert

The variables used within the notification templates are described here 119.

2.3.11 Automatic Links

PeerReview Complete creates links in text that is typed by users as review comments, custom fields, etc... Several types of links are created automatically:

- URLs create links to that URL
- Names of files in the review create links that open that file in the diff viewer

© 2003-2011 SmartBear Software

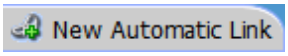
- Line numbers (e.g. 'line 345') in a file comment create links to that line in the diff viewer

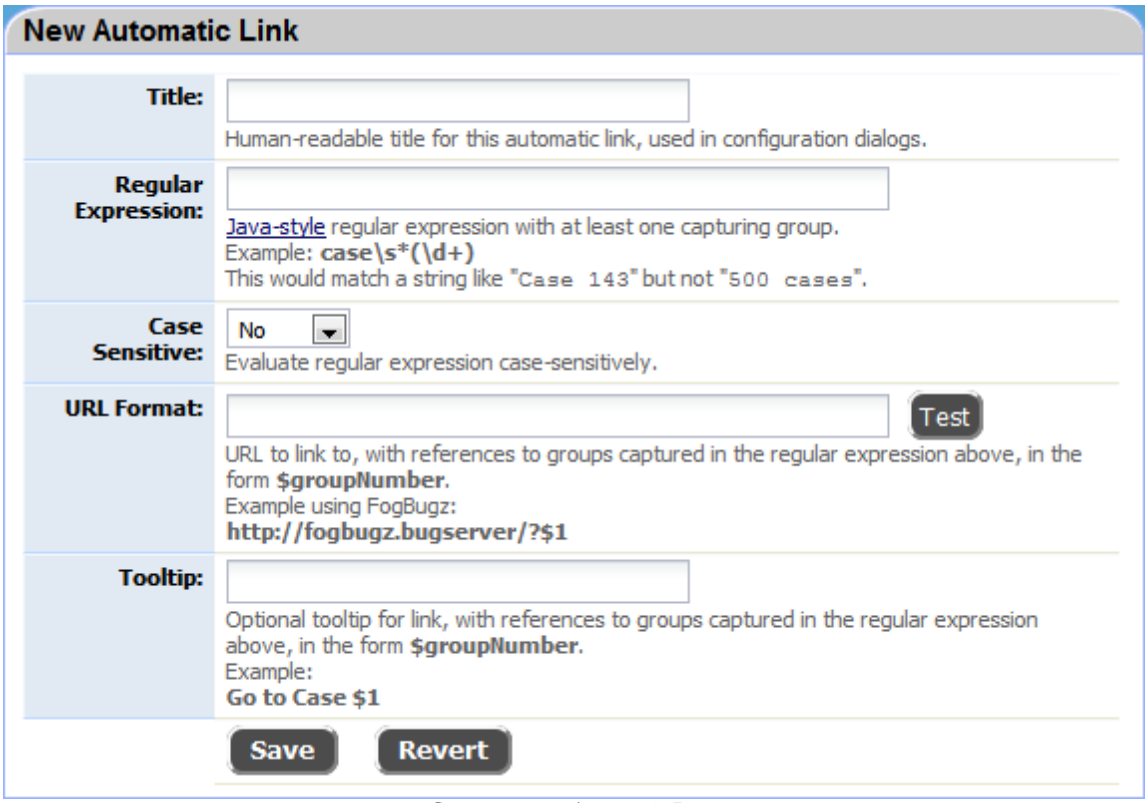
You can define your own Automatic Links to integrate PeerReview Complete with external systems. Each Automatic Link consists of a regular expression pattern, URL format, and optionally a tooltip. When a match is found, PeerReview Complete creates a hyperlink around the matching text. The URL of the hyperlink and the tooltip are constructed by replacing references to groups with the text selected by the corresponding capturing groups in the regular expression.

Examples of external systems you might want to link to include issue/bug trackers, build tools, or documentation library systems.

Creating New Automatic Links

To create a new Automatic Link, click the *New Automatic Link* button.




New Automatic Link
Title:
Human-readable title for this automatic link, used in configuration dialogs.
Regular Expression:
Java-style regular expression with at least one capturing group.
Example: case\s*(\d+)
This would match a string like "Case 143" but not "500 cases".
Case Sensitive:
No
Evaluate regular expression case-sensitively.
URL Format:
Test
URL to link to, with references to groups captured in the regular expression above, in the form $groupNumber.
Example using FogBugz:
http://fogbugz.bugserver/?$1
Tooltip:
Optional tooltip for link, with references to groups captured in the regular expression above, in the form $groupNumber.
Example:
Go to Case $1
Save
Revert

Creating a new Automatic Link

Title — Human-readable title for this automatic link, only used in the Automatic Link Configuration page.

Regular Expression — A Java-style regular expression that identifies a reference to the external tool in text. This is used matched against user-entered text in review titles, custom fields, comments, and defects.

© 2003-2011 SmartBear Software

	You must include at least one one grouping expression (parenthesis). This is combined with the URL Format to link to construct the hyperlink to the external tool. For example: `case\s*(\d+)` This would match a string like `Case 1423` but not `500 cases`. This is the standard way to refer to a bug in FogBugz.
Case Sensitive	Evaluate regular expression case-sensitively. Usually this is set to "no".
URL Format	URL to link to, with references to groups captured in the regular expression above, in the form *$groupNumber*. Example using FogBugz: http://fogbugz.bugserver/?$1
Tooltip	Optional tooltip for link, with references to groups captured in the regular expression above, in the form *$groupNumber*. Example: Go to Case $1

2.3.12 Variable Substitution

In a few areas of server configuration (e.g. Collaborator triggers 115, Perforce triggers 383) you are allowed to use variable substitution to insert useful live data into a text-based template.

This section details those variables and how they work.

Syntax
Variable substitution is indicated with a dollar sign followed by curly braces surrounding the variable name. For example, to access the `review.creator.name` variable you would use this:

```
${review.creator.name}
```

When embedding in other text, variables are inserted into the text wherever they should be replaced. An example that is commonly used in the changelist-update Perforce trigger 383 is:

```
ID ${review.id}: "${review.title}" by ${review.participants.
rolename}
```

Reference
The following table lists all the variable substitutions.

`${review.id}`	The numeric ID of a review. Example: `4235`
`${review.title}`	The title field of a review.
`${review.workflow}`	The name of the workflow this review is currently in. If you change the name in workflow configuration 112 this text changes too, so be careful of scripts which depend on this value.

© 2003-2011 SmartBear Software

`${review.phase}`	The name of the phase this review is currently in, e.g. "Inspection" or "Completed" or "Canceled".
`${review.phase.previous}`	The name of the previous phase of the review. Valid only in the context of a phase change.
`${review.custom.*}`	The value of one of the review custom fields 104. The final variable component should be the full title of the custom field, including whitespace and so forth. If this review doesn't have this field because the user hasn't entered it yet, if there's a default value it will be returned. If this review doesn't have this field because the workflow doesn't include the field, you will get an error message. If you supply the name of a custom field that doesn't exist, you'll get an error message to that effect including the list of valid custom field names. You can use this fact to create an error on purpose to get the list of valid custom fields.
`${review.creator.*}`	User information corresponding to the creator of the review. The asterisk should be replaced by a user-specific variable (described elsewhere). So for example the user's full name would be `${review.creator.name}`.
`${review.author.*}`	User information corresponding to the primary author assigned in the review. The asterisk should be replaced by a user-specific variable (described elsewhere). So for example the user's full name would be `${review.author.name}`.
`${review.participants.rolename}` `${review.participants.rolelogin}`	List of role/user pairs of the people assigned to the review. In the `rolename` case the user's full names are used; in `rolelogin` the user's system login names are used. Roles are named by the display name as configured 107; be careful when writing scripts that depend on this text, and remember that different workflows can define different text for the role name.
`${review.changesummary}`	A multi-line summary of the changes in the review, lines (+add, *mod, -del) for each file.
`${review.shortchangesummary}`	A short summary of the changes in the review, total lines (+add, *mod, -del for all the files in the review.
`${review.defectlog}`	The log of all defects in a review.
`${review.group.title}`	Title of the group associated with the review.
`${review.group.guid}`	GUID of the group associated with the review.
`${user.id}`	The unique ID of the user in our own database.

© 2003-2011 SmartBear Software

`${user.login}`	The system login for the user. If LDAP authentication is being used, this is the user's LDAP login. The login is unique for every user in the system.
`${user.name}`	The user's full name, as they configured it themselves. This is not guaranteed to be unique in the system.
`${user.email}`	The user's email address, or blank if it's not configured.
`${user.phone}`	The user's phone number field, exactly as they configured it themselves.
`${files.paths}`	A set of file paths, in no particular order, separated by the pipe symbol. This is typically a set of paths in the SCM server style, not local file paths.
`${changelist.paths}`	Same as `${files.paths}`.
`${changelist.scmid}`	The ID of this changelist according to the SCM system. For example, Perforce changelist ID or Subversion revision number.
`${changelist.author}`	The author of the changelist according to the SCM system. Note that although this username *might* correspond to a user in PeerReview Complete, this is not required. Further note that because this is a simple te field and not a user object, you cannot continue with other sub-fields like email or phone number.
`${defect.id}`	The unique ID of this defect in our own database.
`${defect.creator.*}`	User information corresponding to the creator of the defect. The asterisk should be replaced by a user-specific variable (described elsewhere). So for example the user's full name would be `${defect.creator.name}`.
`${defect.file}`	The full path to the file this defect is attached to, or a blank string if the defect was review-wide.
`${defect.line}`	The line number in the file where this defect is attached, or 0 if this defect i attached to the overall file, the overall review, or something else that's not relevant to a specific line. Note that this line number refers to the file at the time of review; in the future the file can change and line numbers can change with it.
`${defect.text}`	The text entered in the comment field for this defect. This can be long and multi-lined.
`${defect.isexternal}`	Boolean text "true" if the defect is currently marked external, "false" if the defect is currently open.
`${defect.externalname}`	The external name specified when the defect was marked external.
`${defect.isfixed}`	Boolean text "true" if the defect is currently marked fixed, "false" if the defect is currently open.

© 2003-2011 SmartBear Software

`${defect.custom.*}`	The value of one of the defect custom fields 104. The final variable component should be the full title of the custom field, including whitespace and so forth. If this defect doesn't have this field because the workflow doesn't include the field, you will get an error message. If you supply the name of a custom field that doesn't exist, you'll get an error message to that effect including the list of valid custom field names. You can use this fact to create an error on purpose to get the list of valid custom fields.
`${role.title}`	The title of the role assigned to the user in the review
`${role.description}`	The description of the role assigned to the user in the review
`${actor.*}`	The user responsible for triggering this action. The asterisk should be replaced by a user-specific variable (described elsewhere).

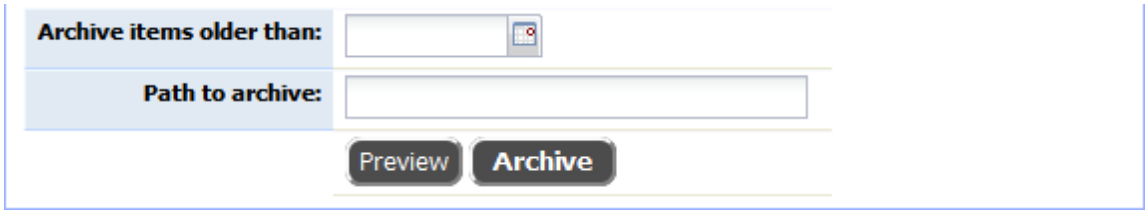
2.3.13 Archiving Data

PeerReview Complete stores the contents of files under review on the server. Over time, this cache will grow to be quite large, and will periodically need to be purged.

Archiving Files
The Archive section of the Admin interface allows you to check the current status of the cache and to archive (or delete) file content that hasn't been archived in a long time.

In the first section of the interface, the total number of files and their collective size is *estimated*. With many thousands of files and gigabytes of data, you don't want to wait for a complete scan of the file system, so this is a ball-park estimate. Use standard file system tools to determine these numbers precisely.

In the second section of the interface you can archive old files:



A file is considered "old" and ready for archiving if its last-used-in-a-review date precedes the user-supplied date. The user interfaces gives a more complete description of the algorithm used to determine this. You must also supply a path in which archive files should be moved.

If you elect to `Preview` the archiving, the system will determine how many files and how much data size will be moved off by the archive.

© 2003-2011 SmartBear Software

Once you select `Archive`, the operation will complete. Once the files are archived, they can be preserved in long-term storage or deleted entirely.

Besides archiving, you can also manually back up [46] or move files out of the cache.

Restoring Archived Files
To restore files after they have been archived, simply copy them back into the live directory.

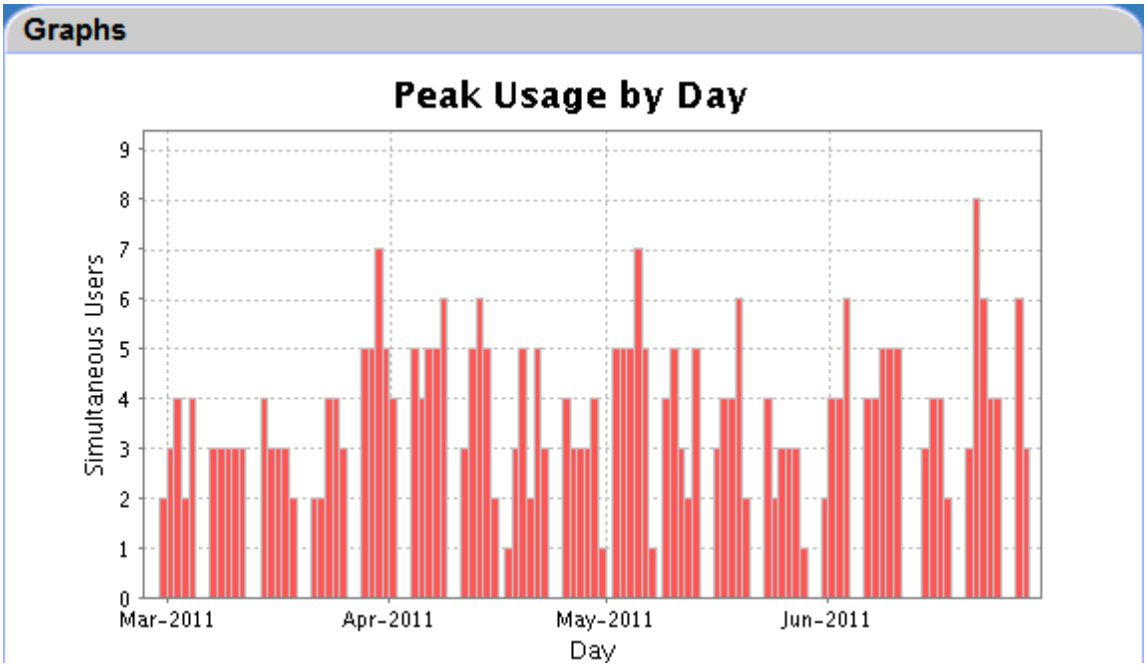
The directory structure must be preserved.

The copy operation can be done while the server is running; users will not be disrupted.

2.3.14 System Status

PeerReview Complete system status reports can be found on the System Status page.

Peak Usage Chart
The "Peak Usage" chart plots the maximum number of simultaneous users, measured by day:



This is most useful when determining the number of floating seats necessary to handle your PeerReview Complete traffic. In the example above, 7 floating seats would probably be sufficient.

Server Backup/Debugging Dump
You can also find the server backup [46] form here.

Many server parameters are displayed. These are typically useful only when debugging problems.

© 2003-2011 SmartBear Software

System Information
Following the debugging forms, charts will display a long list of charts is a long list of system information. These charts will include:

- Debugging (Diagnostic Utility)
- Client/Server Connection
- Database Connection
- Server Properties
- VM Information

These charts will be mainly used to gather system information for our Technical Support when trouble should arise.

3 Web User's Guide

This is the User's Guide for the web server user interface. Installation and administration of the web site is described elsewhere 24. User's Guides for other clients are also available elsewhere.

This guide assumes the web site has already been installed and configured 24 by an administrator.

The web interface supports the following browsers:

- Internet Explorer v6.0, v7.0, v8.0
- Firefox v1.5, v2.0, v3.x
- Safari v3.x and higher
- Chrome v3.x and higher

Other browsers and versions *might* work to varying degrees. Your mileage may vary.

3.1 Account Management

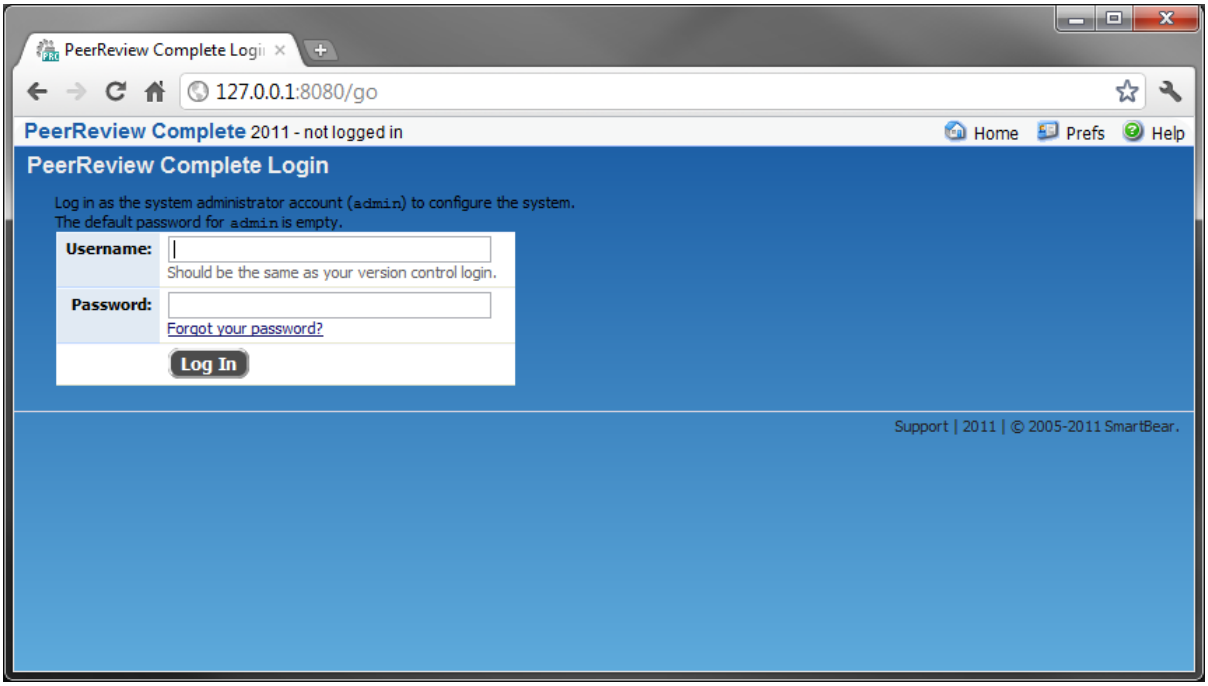
This chapter describes how to log in and out 125 and the options 127 available to you as a user of PeerReview Complete.

It also explains the events upon which you will get a notification, either an email 99 or entry in the RSS feed 135.

© 2003-2011 SmartBear Software

3.1.1 Log in / Log out

When you visit the main PeerReview Complete web page, you are prompted with a log-in screen:



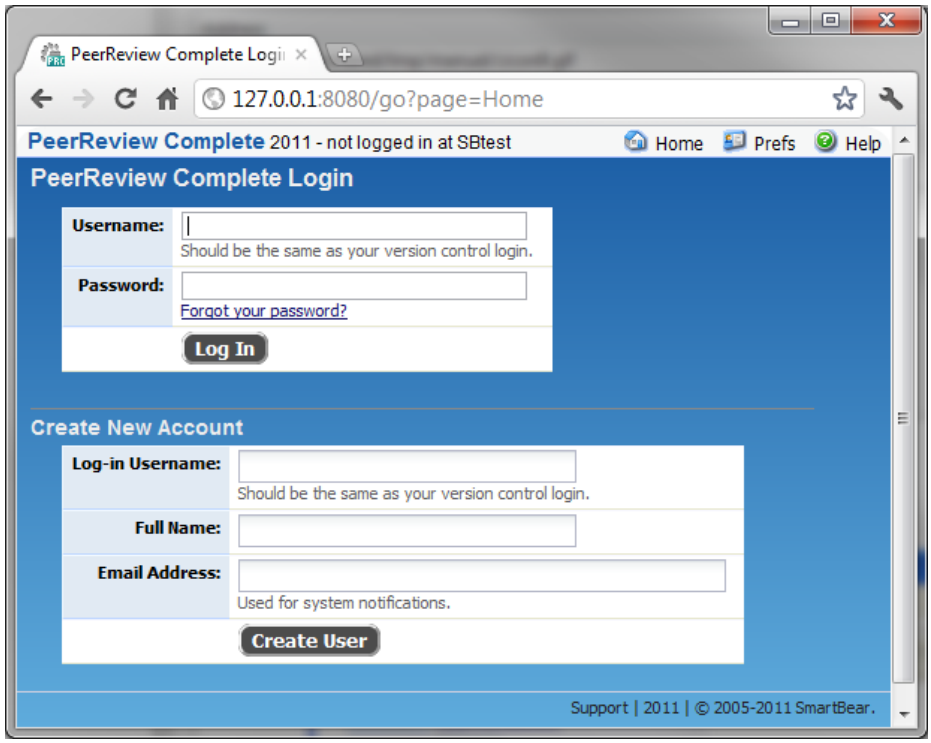
The URL to this screen will depend on your system administrator. It often includes a special port number.

Typically, accounts in PeerReview Complete will be identical to your version control system account.

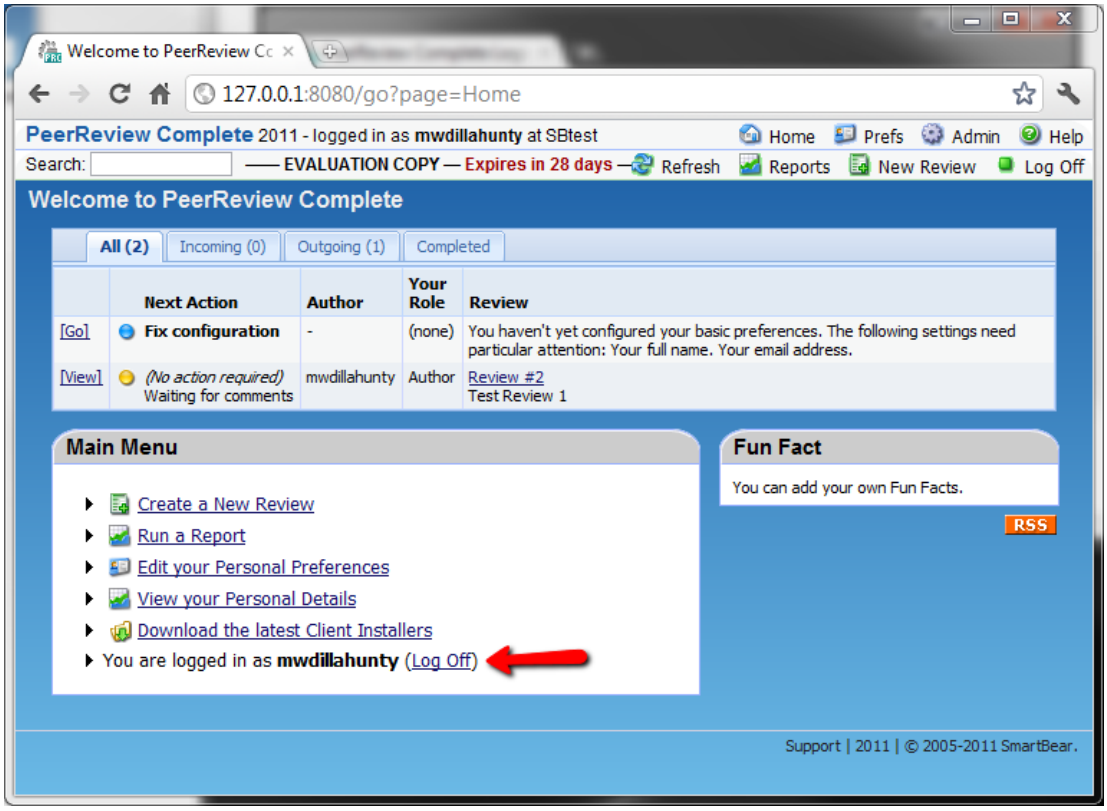
If your account is not set up, you will need to contact an administrator to set one up 84. If you use LDAP or ActiveDirectory 50 for authentication in your company, you can use that username/password with PeerReview Complete and it will automatically create your user account.

© 2003-2011 SmartBear Software

Some system administrators will haveenabled [73] a "Create an account" form on the front page:



Once you're logged in, the system displays this fact at the top of the screen:

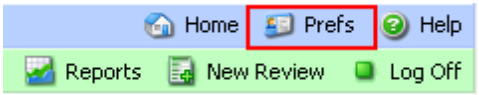

Welcome to PeerReview Cc
127.0.0.1:8080/go?page=Home
PeerReview Complete 2011 - logged in as mwdillahunty at SBtest
Home Prefs Admin Help
Search:
EVALUATION COPY — Expires in 28 days
Refresh Reports New Review Log Off
Welcome to PeerReview Complete
All (2) Incoming (0) Outgoing (1) Completed
Next Action Author Your Role Review
[Go] Fix configuration - (none) You haven't yet configured your basic preferences. The following settings need particular attention: Your full name. Your email address.
[View] (No action required) Waiting for comments mwdillahunty Author Review #2 Test Review 1
Main Menu
Create a New Review
Run a Report
Edit your Personal Preferences
View your Personal Details
Download the latest Client Installers
You are logged in as mwdillahunty (Log Off)
Fun Fact
You can add your own Fun Facts.
RSS
Support | 2011 | © 2005-2011 SmartBear.

You can log out using the link near the bottom of the home page or use the "Log Off" button in the menu bar, but this is usually not necessary.

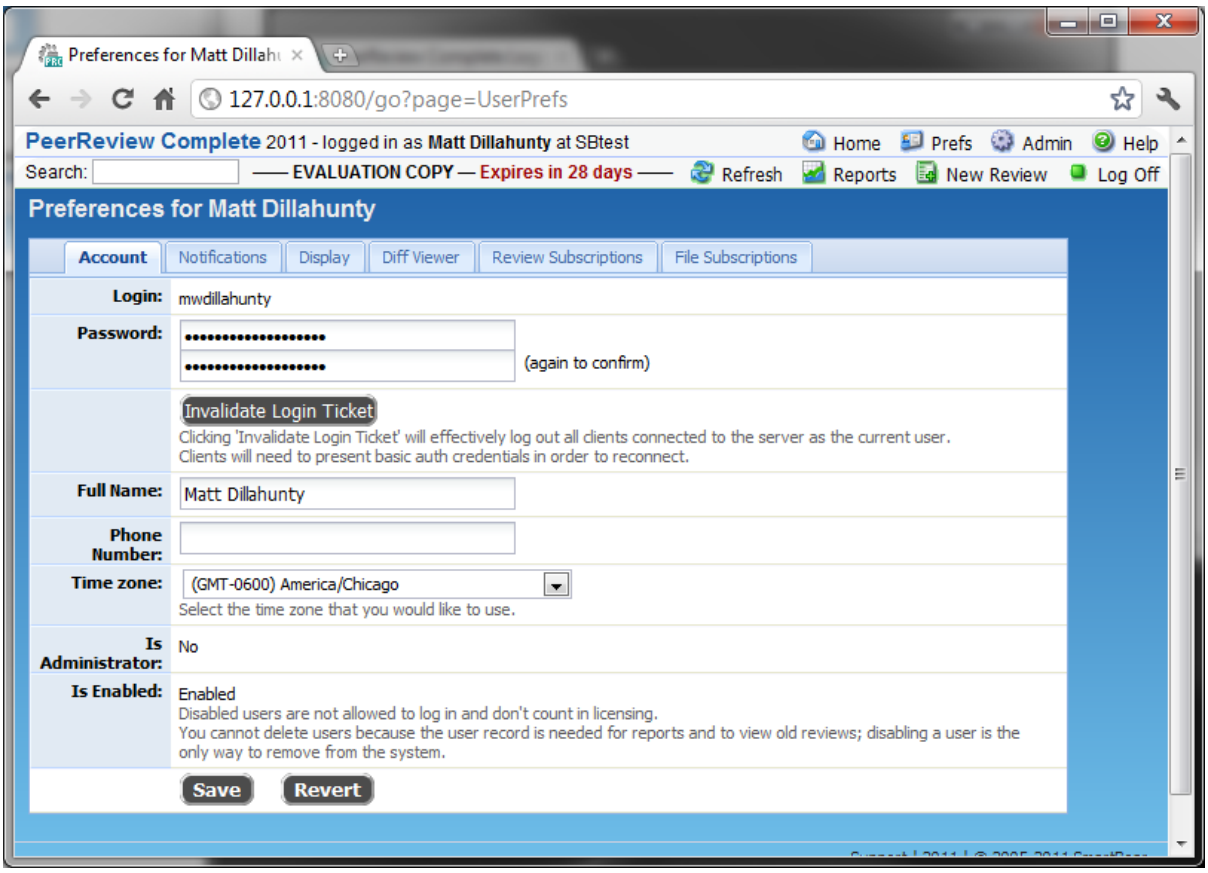
If you don't explicitly log out, the system will give your browser a cookie so the next time you go into a page you will not have to re-login first.

3.1.2 User Preferences

To edit your settings and preferences, use the `Prefs` menubar command:



Account Information



Login

The token you use to log into 125 the system. Typically this will be identical to your version control login name.

Normally, you cannot edit your own login name. However, system administrators are able to edit anyone's login name, so contact an administrator if you need this done.

Password

The password needed to log into the system. Can be blank for "no password," although in all cases the password will be displayed with many dots as shown in the screenshot above. This prevents the causal observer from knowing anything about your password, even its length.

When changing your password you have to type it in twice to confirm.

Full Name

Your full, real name. This is displayed to other users everywhere in the system. You can change this setting at any time.

If you leave this field blank your login name will be used for your name. This is undesirable however; most other users will not know you by your login name.

© 2003-2011 SmartBear Software

Phone Number The phone number other users can reach you at. This field is optional.

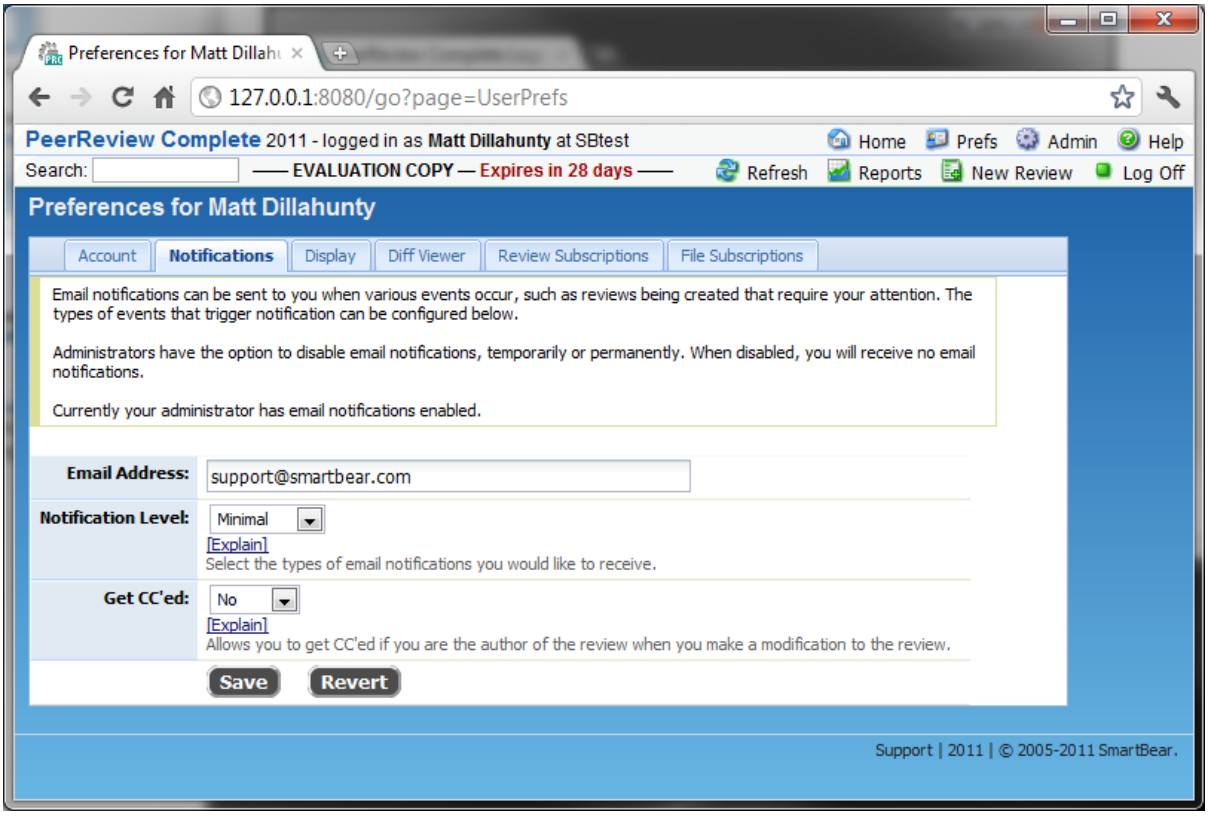
Time Zone Your time zone. This will be used to determine users's displayed date/time

This is defaulted to the time zone selected on the PeerReview Complete server

Is Administrator Controls whether you are a system administrator 87. Typically, you are not able to change this field. Only if you are an administrator yourself can you change this. Be careful: If you remove administrative access from yourself, there's no going back! (Unless another administrator adds you back.)

Is Enabled This field is used by system administrators to determine whether you're allowed 86 to log into the system. If you're looking at your preferences, you can log in, so you'll always see this as "enabled."

Notification Preferences


Preferences for Matt Dillah
127.0.0.1:8080/go?page=UserPrefs
PeerReview Complete 2011 - logged in as Matt Dillahunty at SBtest
Home
Prefs
Admin
Help
Search:
EVALUATION COPY — Expires in 28 days
Refresh
Reports
New Review
Log Off
Preferences for Matt Dillahunty
Account
Notifications
Display
Diff Viewer
Review Subscriptions
File Subscriptions
Email notifications can be sent to you when various events occur, such as reviews being created that require your attention. The types of events that trigger notification can be configured below.
Administrators have the option to disable email notifications, temporarily or permanently. When disabled, you will receive no email notifications.
Currently your administrator has email notifications enabled.
Email Address:
support@smartbear.com
Notification Level:
Minimal
[Explain]
Select the types of email notifications you would like to receive.
Get CC'ed:
No
[Explain]
Allows you to get CC'ed if you are the author of the review when you make a modification to the review.
Save
Revert
Support | 2011 | © 2005-2011 SmartBear.

E-Mail Address The e-mail address that you want all notifications delivered to, and also to display to other users in the system in case they want to contact you directly via email.

© 2003-2011 SmartBear Software

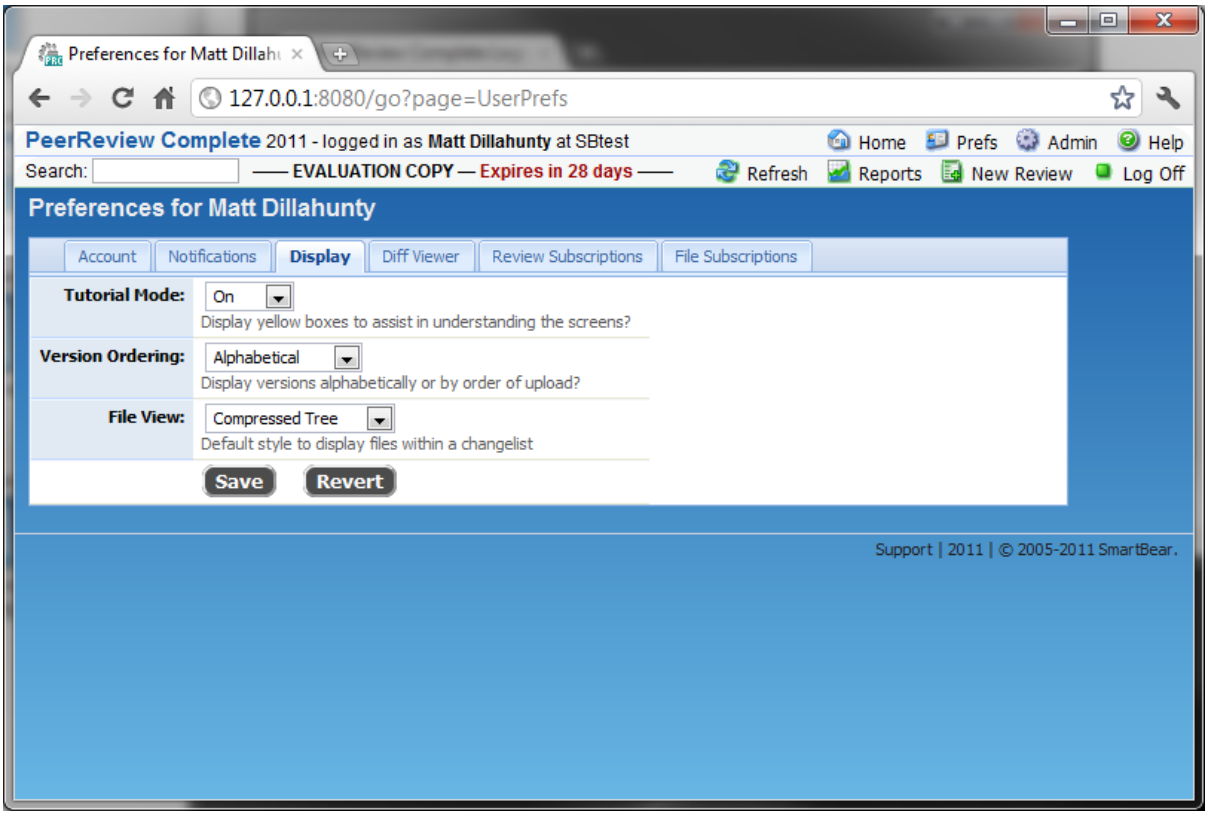
Although this field is technically not required, you really should fill it in, and if you don't, you'll get a warning in your Action Items list 139.

Notification Level This controls how many e-mails you want to get. See the `[Explain]` link for details on the options. Generally, these are the choices:

- Get all notifications, even if you were the cause of the notification event
- Get notification only of those events that you didn't cause yourself
- Get no email notifications

Get CC'ed This allows you to get CC'ed if you are the author of the review when you make a modification to the review.

Display Preferences



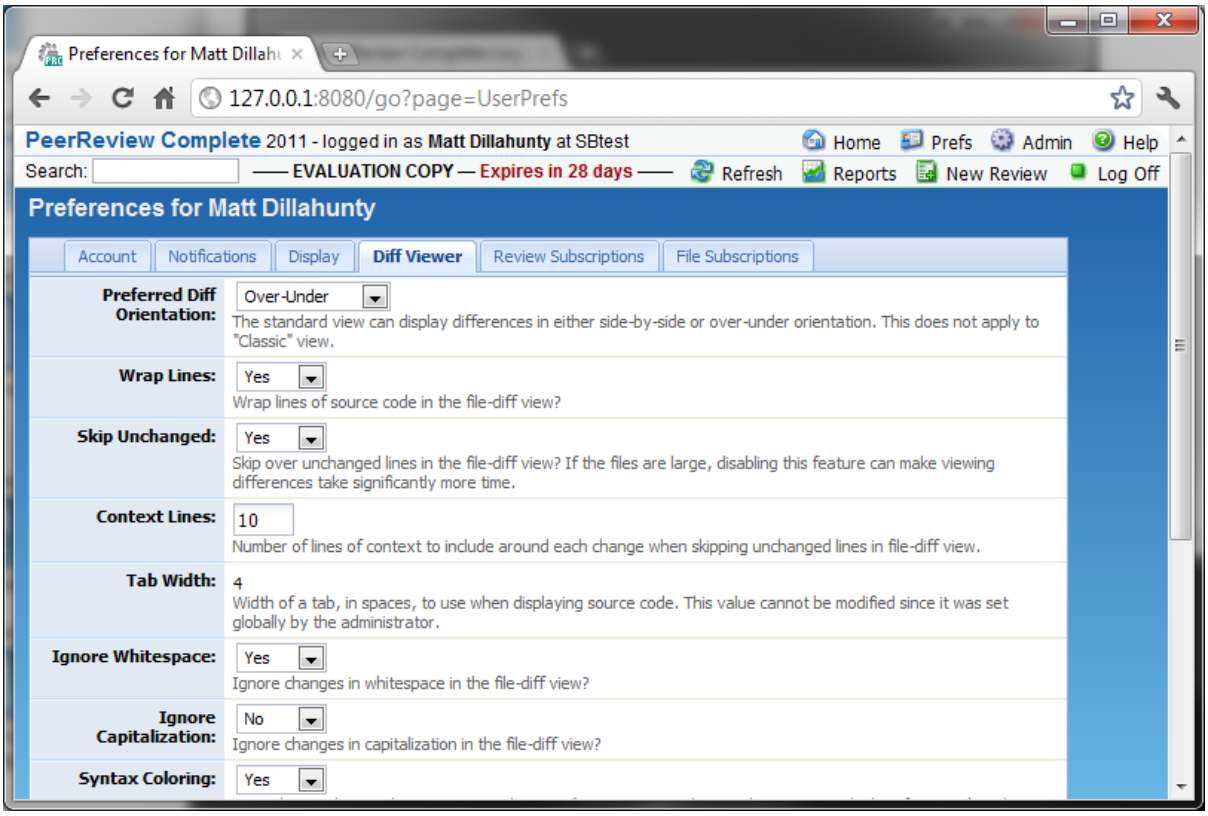
Tutorial Mode Controls whether those little yellow tutorial boxes are displayed throughout the system. By default, these boxes are displayed to help you understand th user interface.

Version Ordering Controls how versions are displayed, alphabetically or by order of upload.

© 2003-2011 SmartBear Software

File View Controls which style should be used to display files within a changelist. Choices include Compressed Tree, Tree, or Flat.

Diff Viewer Preferences


Preferences for Matt Dillah
127.0.0.1:8080/go?page=UserPrefs
PeerReview Complete 2011 - logged in as Matt Dillahunty at SBtest
Home
Prefs
Admin
Help
Search:
EVALUATION COPY — Expires in 28 days
Refresh
Reports
New Review
Log Off
Preferences for Matt Dillahunty
Account
Notifications
Display
Diff Viewer
Review Subscriptions
File Subscriptions
Preferred Diff Orientation:
Over-Under
The standard view can display differences in either side-by-side or over-under orientation. This does not apply to "Classic" view.
Wrap Lines:
Yes
Wrap lines of source code in the file-diff view?
Skip Unchanged:
Yes
Skip over unchanged lines in the file-diff view? If the files are large, disabling this feature can make viewing differences take significantly more time.
Context Lines:
10
Number of lines of context to include around each change when skipping unchanged lines in file-diff view.
Tab Width:
4
Width of a tab, in spaces, to use when displaying source code. This value cannot be modified since it was set globally by the administrator.
Ignore Whitespace:
Yes
Ignore changes in whitespace in the file-diff view?
Ignore Capitalization:
No
Ignore changes in capitalization in the file-diff view?
Syntax Coloring:
Yes

Preferred Diff Orientation 157 Controls the diff orientation. You can choose either a side-by-side orientation or an over-under orientation. This does not apply to "Classic" view.

Wrap Lines Controls whether lines should be word-wrapped when displaying files in the side-by-side view. This is also changeable from inside the side-by-side view 157. If word-wrapping is disabled, the two panes of file differences both have vertical and horizontal scrollbars, and both sets of scrollbars are "tied" together so changing one changes the other.

Skip Unchanged Controls whether large numbers of contiguous, unchanged, uncommented lines should be displayed in the side-by-side view. This is also changeable from inside the side-by-side view.

This mode is useful when a large file has a small number of changes but sometimes need to be disabled when you need to look at more of the context in the file.

File content will load much faster in the browser if this feature is enabled since much les text needs to be transmitted from the server to the client.

© 2003-2011 SmartBear Software

Context Lines

If "Skip Unchanged" is true, this controls the number of lines around each change or comment to display before displaying a "skipping lines" message.

Ignore Whitespace

Controls whether changes in whitespace alone should be considered when showing differences in the side-by-side view. This is also changeable from inside the side-by-side view.

Ignore Capitalization

Controls whether changes in capitalization alone should be considered when showing differences in the side-by-side view. This is also changeable from inside the side-by-side view.

Syntax Coloring 161

Controls whether to use color and style to distinguish various programming elements (e.g. comments, keywords, constants, built-in functions, types) when displaying source in the side-by-side view. This is also changeable from inside the side-by-side view.

Disabling coloring can speed up the processing and downloading of very large files. Thi is usually necessary only when running with a slow connection to the server.

Side-by-Side Display Order

Controls whether to display the newer file on the left or the right in a side-by-side difference view.

Default Version Comparison

Controls the default difference view when the file in question has multiple uploaded versions. The "current" version is always displayed; this lets you set the default comparison version to be "the last version that was checked into version control" or "th last version that was uploaded to this review" or "the last version that I pressed the 'Accept' button in the general comment area of".

Font in Side-by-Side

Controls the font family and size for the side-by-side file viewer. This must be in the following format:

```
SIZEpt FONT NAME
```

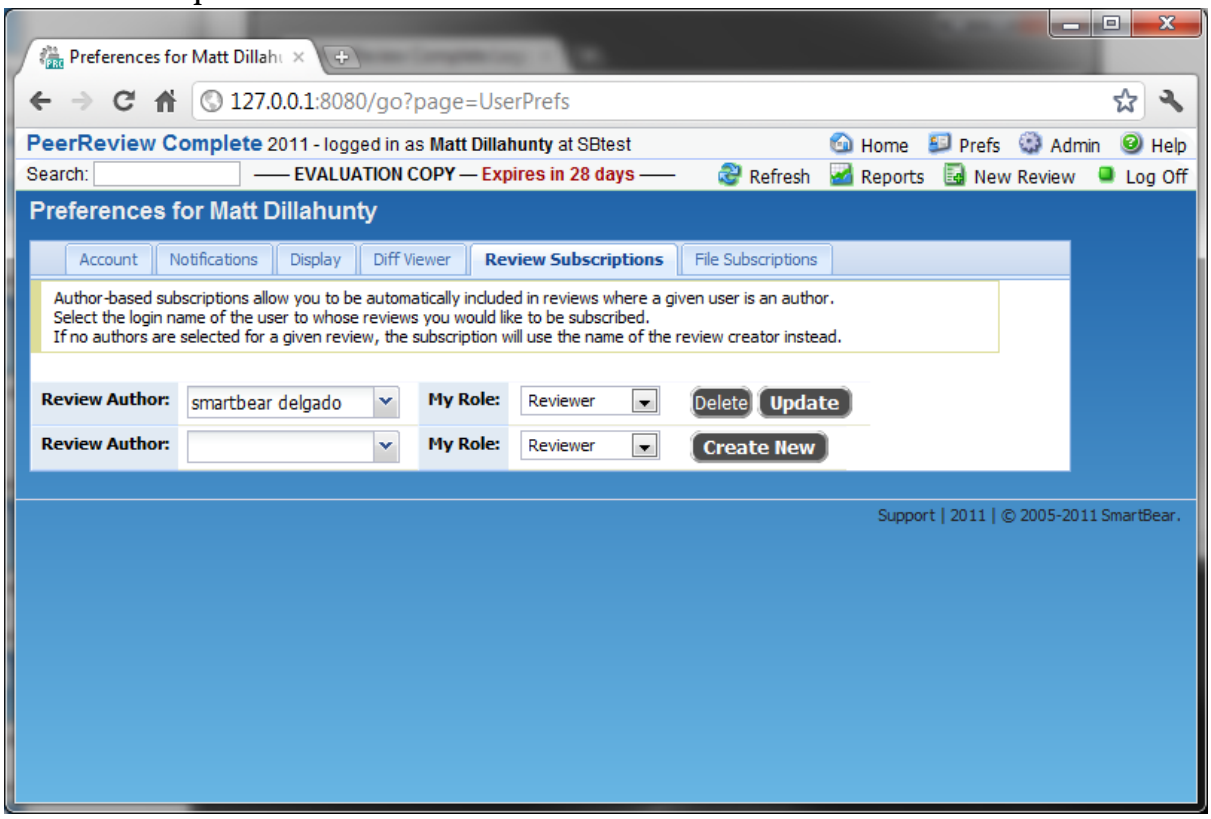
Some examples:

```
9pt Courier New
8pt Tahoma
```

At the bottom of the field the system will preview the current font selection. If you've made an error in the format or font name, this preview will take on the web browser default font and size.

© 2003-2011 SmartBear Software

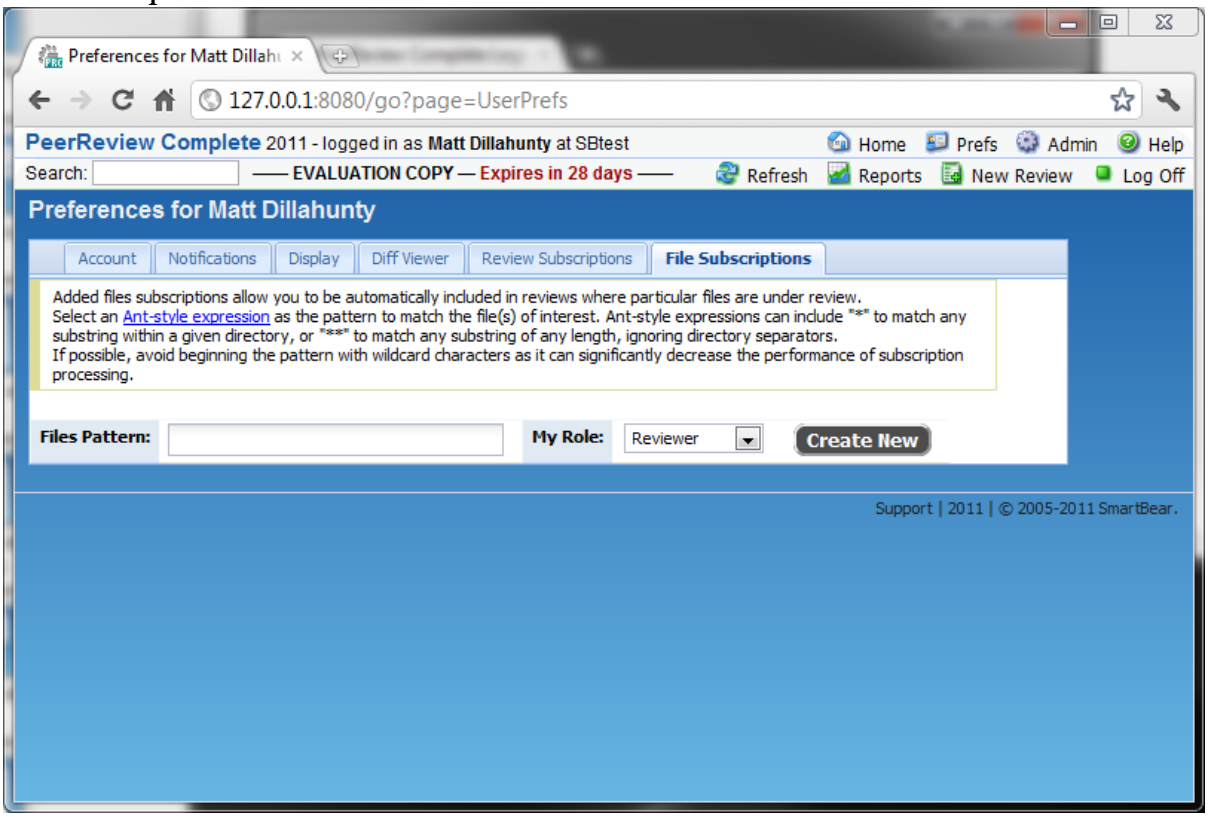
Review Subscriptions Preferences



Review Subscriptions allow you to automatically subscribe and be included in reviews with a preferred author. Under the "review creator" field, select the login name of the user to whose reviews you would like to be subscribed. If no authors are selected for a given review, the subscription will use the name of the review creator instead. You can also choose whether you'd like to be **Reviewer**, **Observer**, or **Moderator** in the review.

© 2003-2011 SmartBear Software

File Subscriptions Preferences


Preferences for Matt Dillah
127.0.0.1:8080/go?page=UserPrefs
PeerReview Complete 2011 - logged in as Matt Dillahunty at SBtest
Home
Prefs
Admin
Help
Search:
EVALUATION COPY — Expires in 28 days
Refresh
Reports
New Review
Log Off
Preferences for Matt Dillahunty
Account
Notifications
Display
Diff Viewer
Review Subscriptions
File Subscriptions
Added files subscriptions allow you to be automatically included in reviews where particular files are under review.
Select an Ant-style expression as the pattern to match the file(s) of interest. Ant-style expressions can include "*" to match any substring within a given directory, or "**" to match any substring of any length, ignoring directory separators.
If possible, avoid beginning the pattern with wildcard characters as it can significantly decrease the performance of subscription processing.
Files Pattern:
My Role:
Reviewer
Create New
Support | 2011 | © 2005-2011 SmartBear.

File Subscriptions allow you to automatically subscribe and be included in reviewers where a particular file is under a review. Like Review Subscriptions 133, you can choose your role as a **Reviewer**, **Observer**, or **Moderator**. Select an Ant-style expression as the pattern to match the file(s) of interest. The expression will be matched to the file path alone; it will not look for matches in the complete repository path with URL and other server descriptions. Ant-style expressions can include "*" to match any substring within a given directory, or "**" to match any substring of any length, ignoring directory separators. If possible, avoid beginning the pattern with wildcard characters as it can significantly decrease the performance of subscription processing.

3.1.3 Notifications

User notifications can come by email 99 or RSS feed 135. When are those notifications sent?

1. To reviewers and observers when the review enters the inspection phase from planning. This happens when the author/review creator clicks the "Move ahead to the inspection" button on the last page of the wizard.
2. To author/review creator when the first comment is made or when the first defect is created.
3. To author/review creator when the review enters the rework phase. Usually this is when the last reviewer clicks the "Review Finished" button on the review overview and there are outstanding defects. The rules for this transition are customizable, but regardless of the rules, it is at this transition that it happens.
4. To reviewers and observers when the review enters the inspection phase from rework. This happens when the author clicks the "Proceed to verification" button on the review overview screen.

© 2003-2011 SmartBear Software

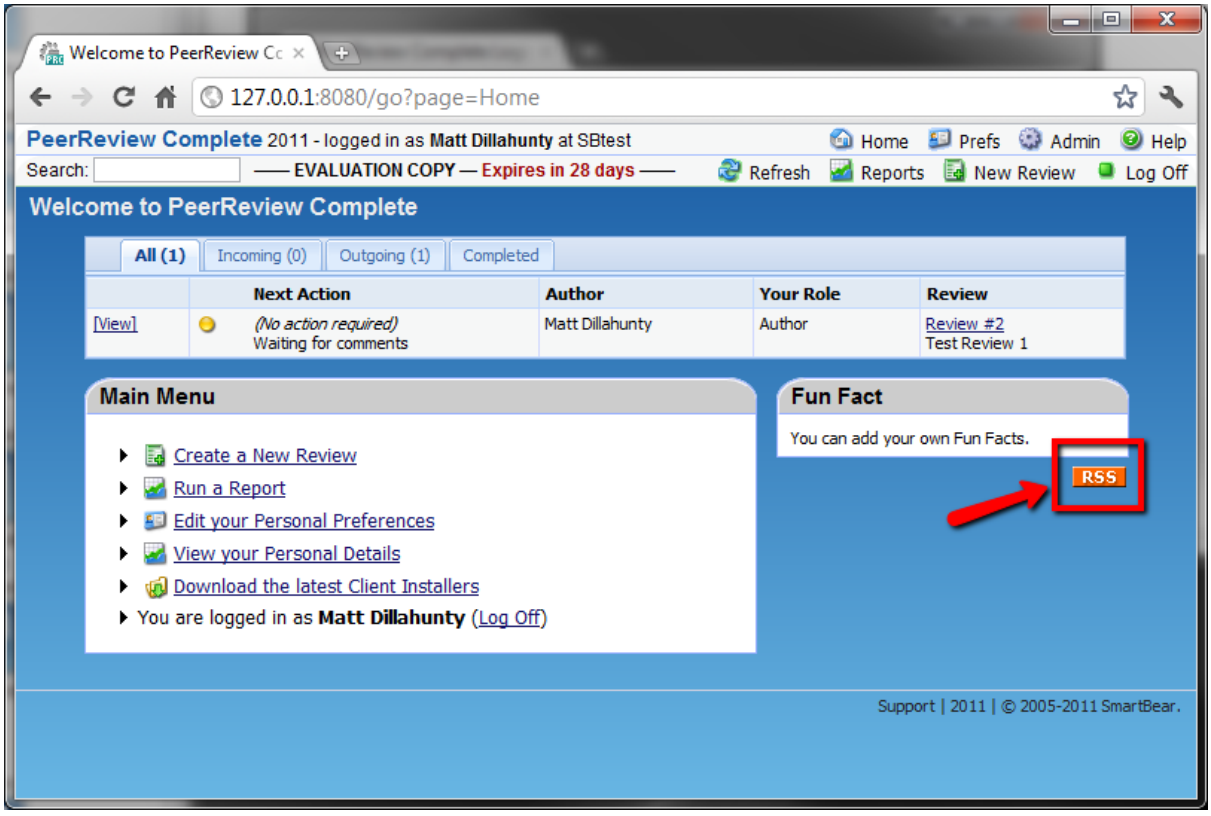
5. To reviewers who have marked the review finished if and when additional discussion takes place.
6. To author when the review is "finished", which happens when the last reviewer is finished and there are no outstanding defects.
7. To all participants when a review is canceled. (NOTE: There is a bug where this email goes out even if the review was only in planning phase and the reviewers have not been notified of the review yet.)

The exceptions to these rules are as follows:

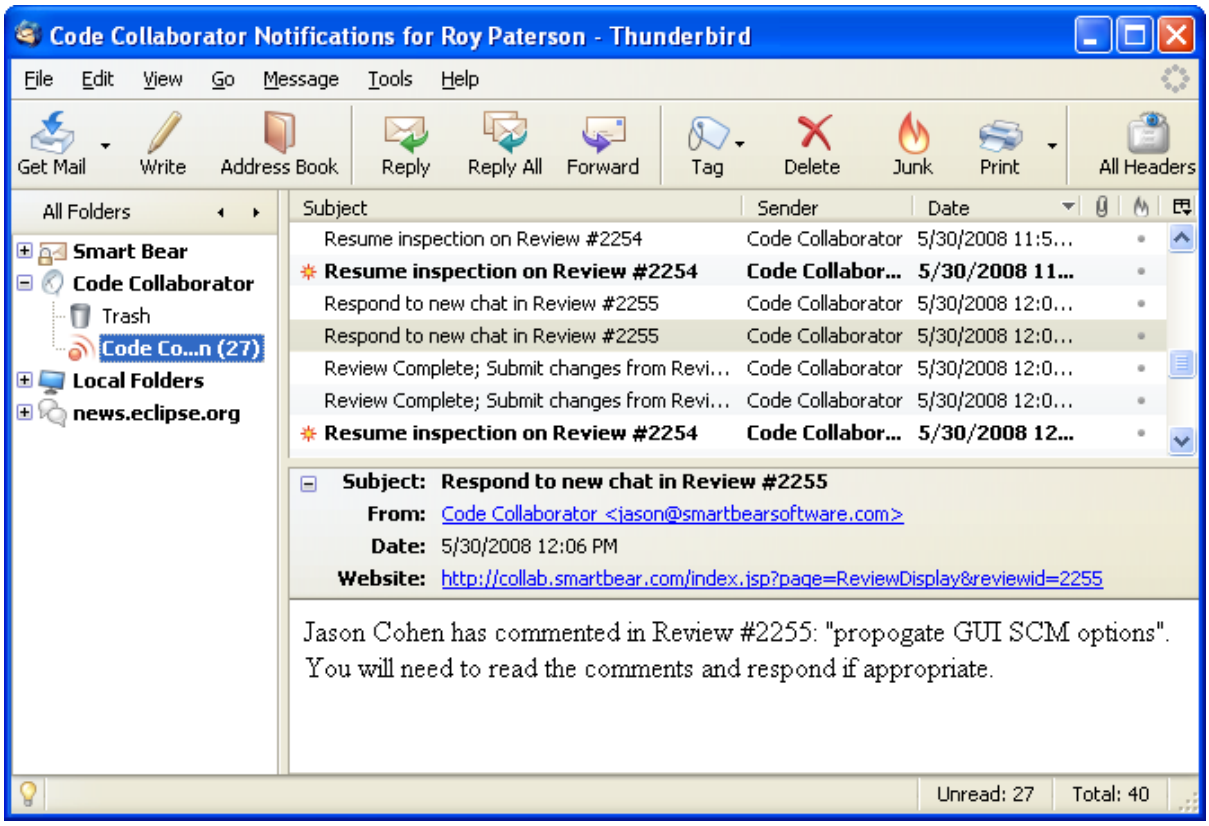
1. If a user has set a user preference to not be notified via email, they will not receive an email.
2. If the reason for the email can be traced back to an action taken by a specific user, that user will not receive an email unless they have changed their preference to request *all* emails. For example, the person canceling a review does not receive the email notification for the cancel action.

RSS Feed

Use the orange button on the bottom right of the home page to configure your RSS reader. The configuration will differ according to which RSS reader you choose to use.



Here is an example of the RSS feed in Mozilla Thunderbird:


Code Collaborator Notifications for Roy Paterson - Thunderbird
File Edit View Go Message Tools Help
Get Mail Write Address Book Reply Reply All Forward Tag Delete Junk Print All Headers
All Folders
Smart Bear
Code Collaborator
Trash
Code Co...n (27)
Local Folders
news.eclipse.org
Subject Sender Date
Resume inspection on Review #2254 Code Collaborator 5/30/2008 11:5...
Resume inspection on Review #2254 Code Collabor... 5/30/2008 11...
Respond to new chat in Review #2255 Code Collaborator 5/30/2008 12:0...
Respond to new chat in Review #2255 Code Collaborator 5/30/2008 12:0...
Review Complete; Submit changes from Revi... Code Collaborator 5/30/2008 12:0...
Review Complete; Submit changes from Revi... Code Collaborator 5/30/2008 12:0...
Resume inspection on Review #2254 Code Collabor... 5/30/2008 12...
Subject: Respond to new chat in Review #2255
From: Code Collaborator <jason@smartbearsoftware.com>
Date: 5/30/2008 12:06 PM
Website: http://collab.smartbear.com/index.jsp?page=ReviewDisplay&reviewid=2255
Jason Cohen has commented in Review #2255: "propogate GUI SCM options".
You will need to read the comments and respond if appropriate.
Unread: 27 Total: 40

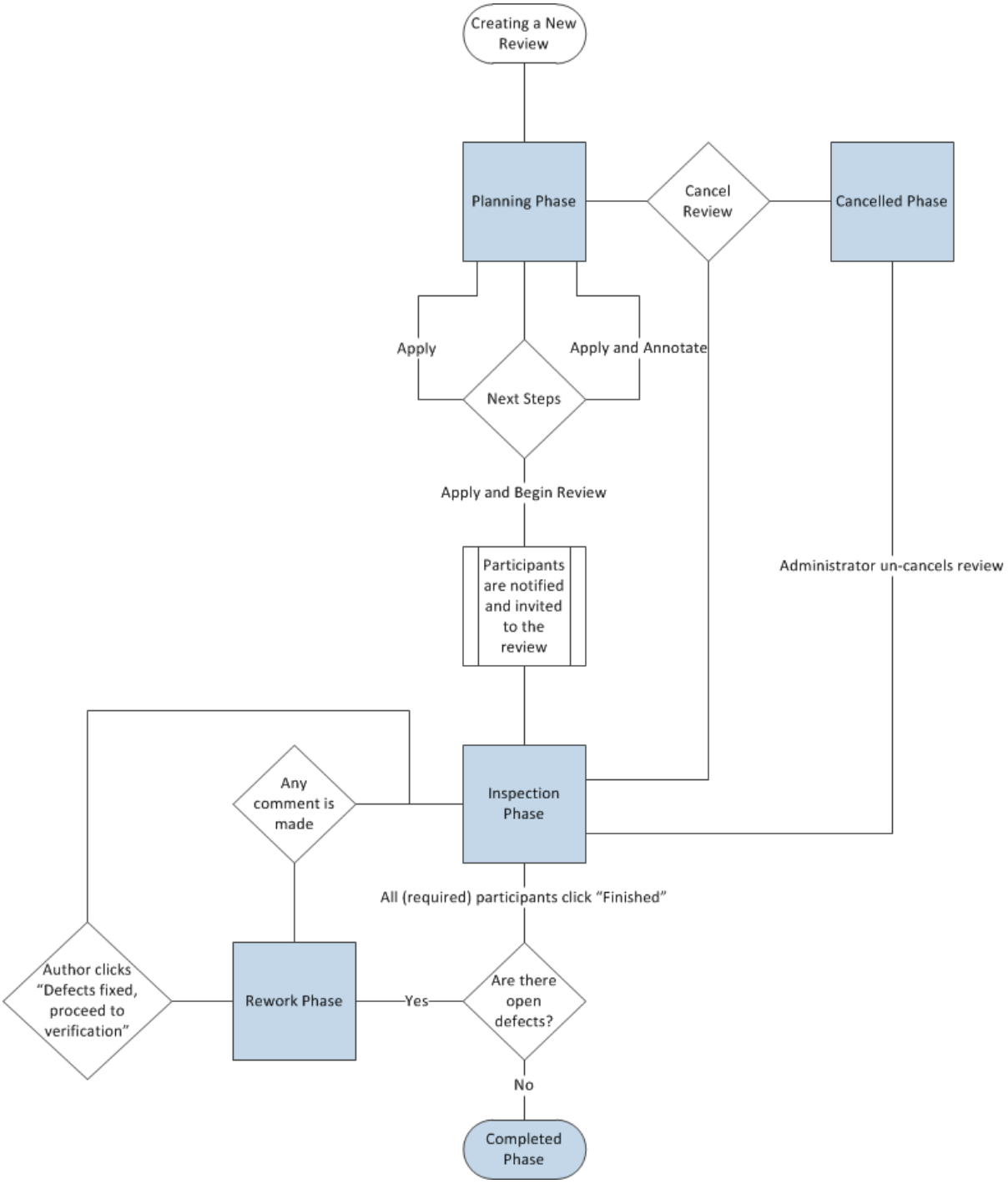
3.2 Reviews

This chapter describes creating, performing, and managing peer reviews.

3.2.1 Review Overview

This section describes a typical review and its phases. The following flowchart guides you through PeerReview Complete's workflow. Review phases are indicated by the blue items.

© 2003-2011 SmartBear Software



Creating a Review

Any of the following actions will create a review:

- Clicking the "New Review" menu option 140.
- Adding new materials to a new review through the GUI Client 189, Command Line Client 196, Eclipse Plug-in, Visual Studio Add-in 286, Perforce Plugins 381, or the Print To Review 282.
- Using the ccollab admin review create command 240 in the Command Line Client.

© 2003-2011 SmartBear Software

- Triggering a script 443 that creates a new review.

Planning Phase
In the Planning phase, only the review creator is active to customize the review using the Review Creation Wizard 140. Here the creator enters review attributes including the title, custom fields 104, attaches review materials 143, and assigns participants 142 to the review.

Using the Next Steps 146 options, you can save your current information and continue in the Planning phase by choosing "Apply" or "Apply and Annotate." Clicking "Apply and Begin Review" will notify the specified participants and move the review to the next phase.

Inspection Phase
Once the review has begun, PeerReview Complete activates all roles and notifies the participants that their actions are required in the review. Participants are presented 148 with the files, file-differences 160, and other review material. Everyone can make comments on individual lines of a file or on entire documents.

Comments 168 work a bit like "instant message" chat and a bit like "newsgroups." If everyone is chatting at the same time, you have a real-time "instant message" environment so the review can progress swiftly. If one or more participants are separated by many time zones or just aren't currently at the computer, the chat looks like a newsgroup where you post comments and receive e-mails when someone responds. This means PeerReview Complete works equally well no matter where your developers or reviewers are located.

If reviewers find problems, they open defects 170 right from the web interface, associated with the file and line number if applicable. Defects are tracked through PeerReview Complete and can optionally be mirrored 171 into an external issue-tracking system.

Once all required 111 participants to indicate that they are finished 156 with the current phase, the review moves to the next phase, which will depend on whether open defects 170 remain in the review. If there are no defects or all defects have been marked external 171 or fixed 171, the review moves to the Completed phase. If the review contains any open defects, the review moves on to the Rework phase.

Rework or Fix Defects Phase
In the Rework phase, the authors are the only roles active so that they may fix the defects found in the Inspection phase. When the authors are satisfied with their fixes, the fixes are uploaded into the review. The authors then indicate they are ready for the fixes to be validated by clicking the "Defects fixed, proceed to verification" button in the Moving On 156 section of the Review Summary page.

For validation, the review moves back to the Inspection phase, where reviewers have access to the new changes as well as the original changes and comments.

If any participant make a comment while the review is in Rework phase, the review will automatically revert back to the Inspection phase in order for additional discussions to take place.

Completed Phase
Only when no open defects remain and all participants required to click "Finished" have done so in the Inspection phase will the review move to the Completed phase.

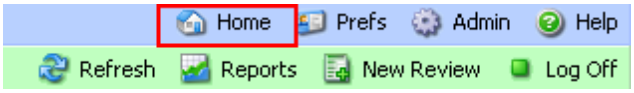
© 2003-2011 SmartBear Software

Cancelled Phase

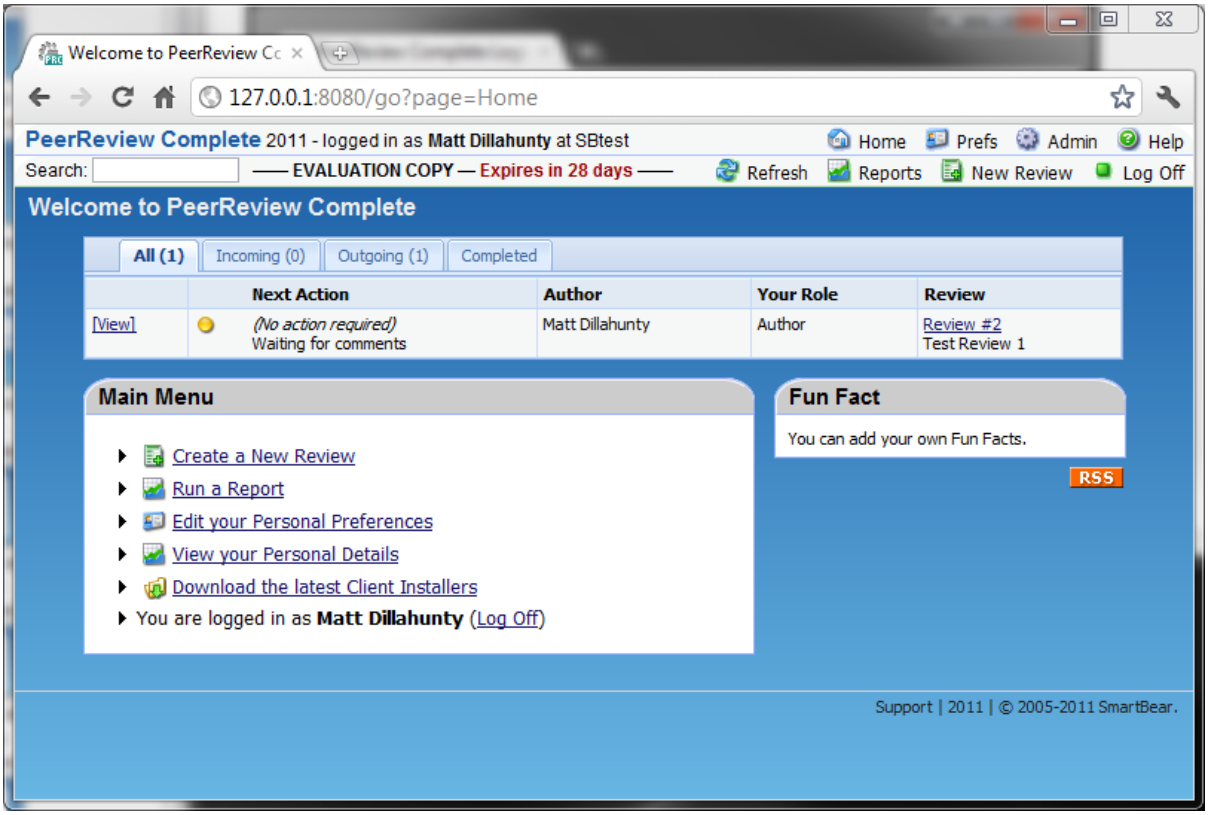
In the Planning and Inspection phases, there may be an option to cancel [148] the review depending on the administrative settings [79]. If this button is applied, the review will move to a Cancelled phase. A cancelled review can only be re-activated by an administrator.

3.2.2 Action Items

Action Items are relevant, current things that you are included in. You can view your "Action Items" on your homepage by clicking the "Home" button on the top right of your screen:



You can also sort action items by clicking on the tabs next to the "All" tab. When the "Incoming" tab is active, you will view all reviews in which you are a reviewer or an observer. The "Outgoing" tab will show all reviews in which you are the author. "Completed" will list recently completed reviews.



Your Action Item list is updated immediately if anything changes. You can refresh this screen manually; the home page refreshes automatically every five minutes.

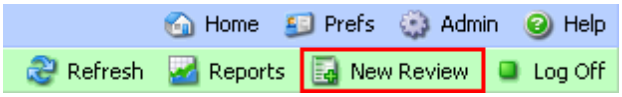
When you have an Action Item directing you to commit files, you remove it by clicking the "dismiss" link on the left. PeerReview Complete uses these Action Items to track whether review materials have been submitted to version control. If PeerReview Complete does not have enough information to determine that the materials are already versioned it will create a reminder for you when the review is completed.

© 2003-2011 SmartBear Software

Action Items are also visible in other clients, but that is covered in the separate User Guides.

3.2.3 Creating a Review

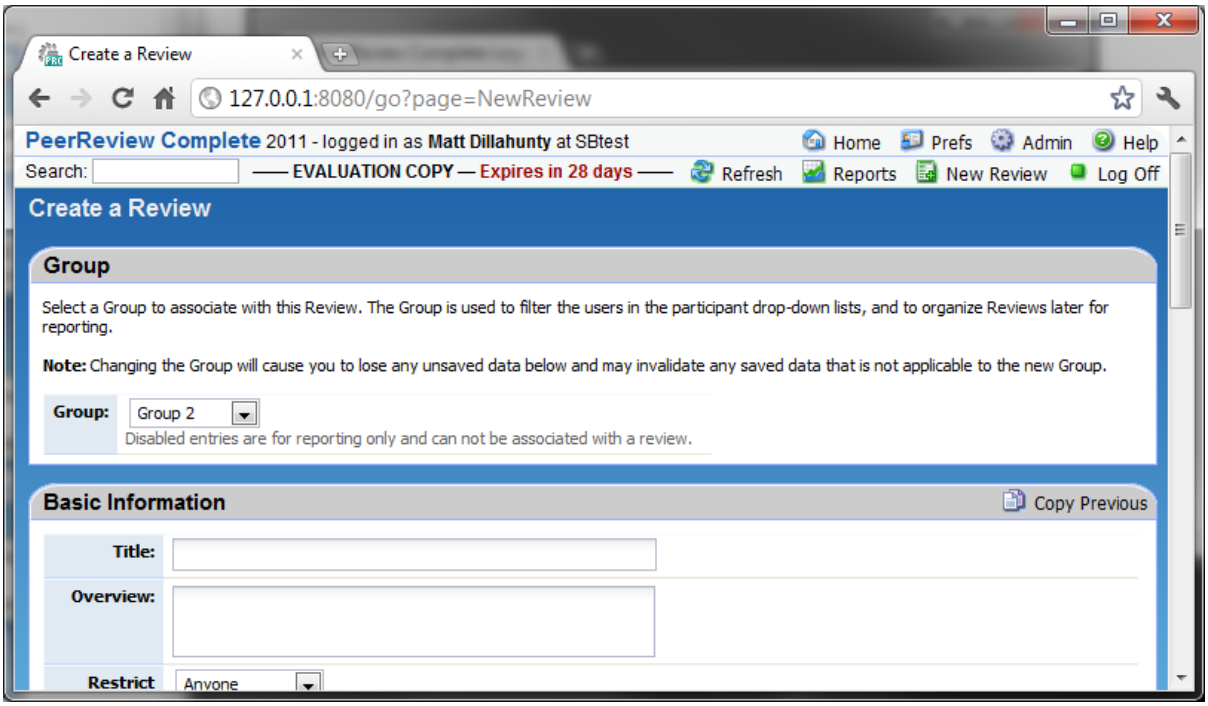
To create a new review, click the "New Review" toolbar button:



You can also click on the "New Review" button on the main menu on the "Home" page.

Creating a New Review
You will see a form in the Review Creation Wizard:

Group
If the Groups feature 88 is enabled and you belong to a group, the first box you will see is the Group box in the Review Creation Wizard:


Create a Review
127.0.0.1:8080/go?page=NewReview
PeerReview Complete 2011 - logged in as Matt Dillahunty at SBtest
Home
Prefs
Admin
Help
Search:
EVALUATION COPY — Expires in 28 days
Refresh
Reports
New Review
Log Off
Create a Review
Group
Select a Group to associate with this Review. The Group is used to filter the users in the participant drop-down lists, and to organize Reviews later for reporting.
Note: Changing the Group will cause you to lose any unsaved data below and may invalidate any saved data that is not applicable to the new Group.
Group:
Group 2
Disabled entries are for reporting only and can not be associated with a review.
Basic Information
Copy Previous
Title:
Overview:
Restrict
Anyone

A **group** must be selected to create the review. The text you see may differ if your system administrator has configured the Group Prompt 75 differently. A drop-down list will also appear with every group you belong to, and you will be able to choose which group to associate with the review using the combo-box.

Once you select a group, the participants list will be updated accordingly to show only those users belonging to the selected group. Changing the group will cause you to lose any unsaved data applied to the review and may invalidate any saved data that is not applicable to the new group.

© 2003-2011 SmartBear Software

If you belong to only one group, you will *not* be given the option of selecting a group; the group will be applied automatically, and the participants list will be narrowed according to the group members.

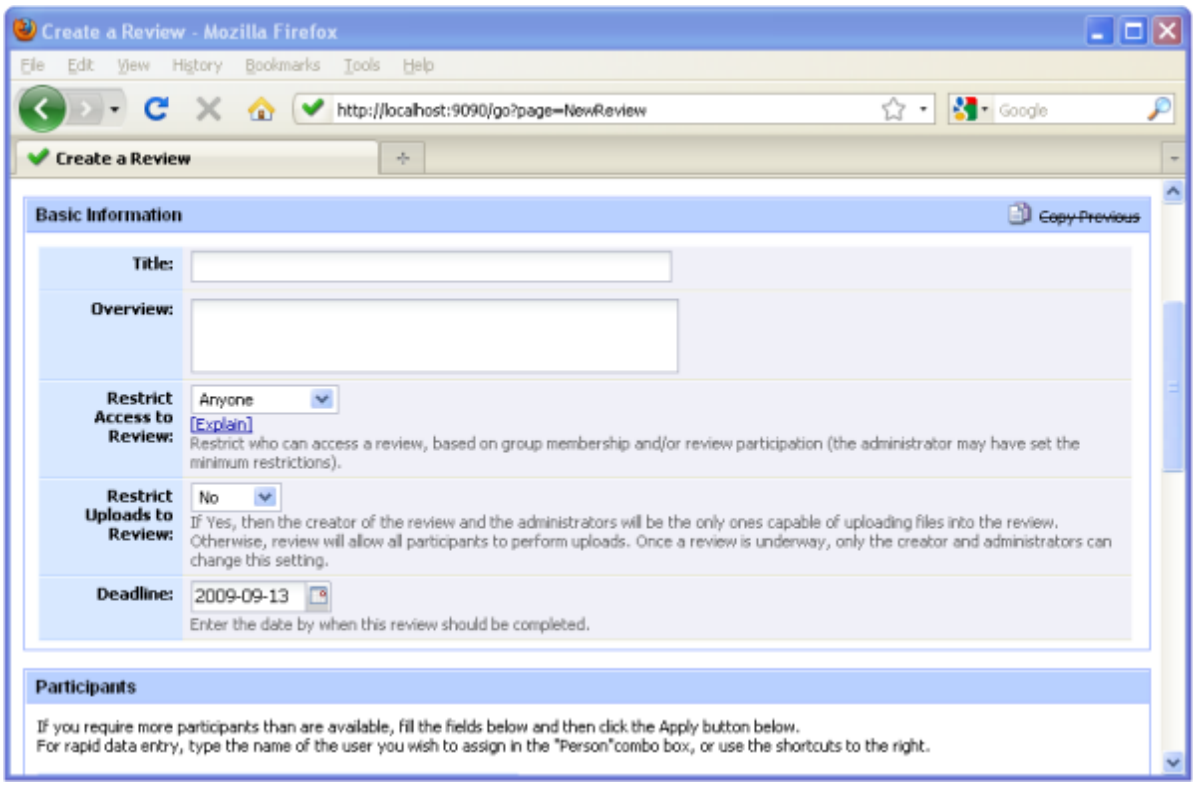
Template

If you have multiple templates configured under "Review Templates," you will see the Template box (shown at the bottom of the previous screenshot).

You must select the **template** for the review. The template defines all the rules and workflow that governs the review. This includes custom fields for reviews and defects as well as the set of roles and rules for participants in the review. Your system administrator will have already configured 112 the templates you can pick from and probably has some guidelines as to which templates should be used under which circumstances.

If there is only one configured template, you will *not* be given the option of selecting a template; the single template will be applied automatically. In that case, the first box shown is "Basic Information."

Basic Information


Create a Review - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:9090/go?page=NewReview
Google
Create a Review
Basic Information
Copy Previous
Title:
Overview:
Restrict Access to Review:
Anyone
[Explain]
Restrict who can access a review, based on group membership and/or review participation (the administrator may have set the minimum restrictions).
Restrict Uploads to Review:
No
If Yes, then the creator of the review and the administrators will be the only ones capable of uploading files into the review. Otherwise, review will allow all participants to perform uploads. Once a review is underway, only the creator and administrators can change this setting.
Deadline:
2009-09-13
Enter the date by when this review should be completed.
Participants
If you require more participants than are available, fill the fields below and then click the Apply button below.
For rapid data entry, type the name of the user you wish to assign in the "Person" combo box, or use the shortcuts to the right.

A **title** must be assigned for the review. The title is used all over the place -- in Action Items 139, in notification emails, to web page titles, and so forth. Make the title brief but descriptive. Titles do not have to be unique in the system; reviews already have a unique ID that is assigned by the server. (In the example given above, the unique ID is "12" and identified by the red circle.)

The next field will be the **overview** field. Here, you can enter anything to describe the review in more detail. This field is optional and does not have to be filled out if you'd like to leave it empty.

© 2003-2011 SmartBear Software

Restrict access to the review allows you to limit those users who may access your review. By default, any user in the system can view and comment on any review at any time. However, there are circumstances where you may want to restrict viewing to those users specifically invited to the review, and this option enables you to do so. Please note that system administrators are able to view all reviews regardless of restricted access selections, and also have the ability to require 73 that all reviews be restricted. In such cases, you will not be given an option to restrict access.

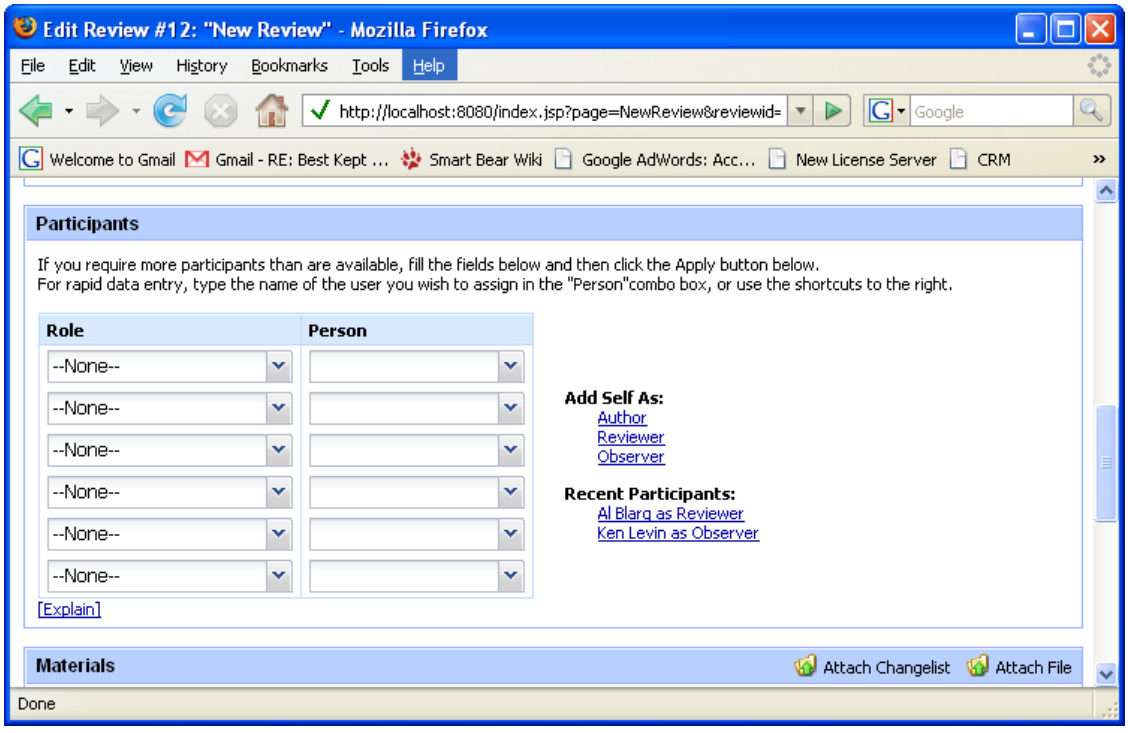
You can also **restrict uploads** to the review by limiting who can upload files into the review. If this setting is set to "Yes," only the creator and the administrators will be allowed to upload files.

Review Deadline is an option that only appears if a system administrator has set the default review deadline to a value other than 0. In that case, you will see the Deadline option with a default date that is calculated based on the value provided by the administrator. You can set the review deadline to any date that is on or after the current date. On the day before the deadline, participants will see a note in their action items that that review is 'Due today'. This is intended to alert participants that if the review is not completed by midnight of the current day, it is overdue.

If you have previously created reviews, you can also choose the same custom field setting as your previous review by clicking the **`Copy Previous`** button next to the Delete Review 148 option.

Review Participants

The next step in creating a review is to invite people to the review, and give each of them a role to play.



Since the administrator has complete control 107 over the number of roles and their definitions, it's difficult to give specific information about them.

Some examples of commonly-used roles are as follows:

© 2003-2011 SmartBear Software

- Author: Responsible for the files under review and probably responsible for fixing any defects that are found.
- Reviewer: Responsible for finding defects; is allowed to mark defects "fixed;" the review isn't finished until all reviewers say it's finished.
- Observer: Invited to the review but if all other participants are finished the review is complete. Useful when this user isn't available just now and you don't want to wait for him to show up.
- Moderator: Controls the pace and flow of the review; coordinates other people and decides when the review changes phases.

Each invitee can be assigned *only one* role. However, there is no limit on the number of invites, and, unless the administrator has configured 107 things otherwise, there can be more than one user per role.

You can select each user and role individually through the drop-down menu, or start typing the name of the user in the appropriate field to narrow down the user list. User/role combinations you've used recently will also appear under "Recent Participants." This makes it easy and fast to select common combinations.

Attaching Review Materials

Any number and type of file can be uploaded into the review. This includes but is not limited to:

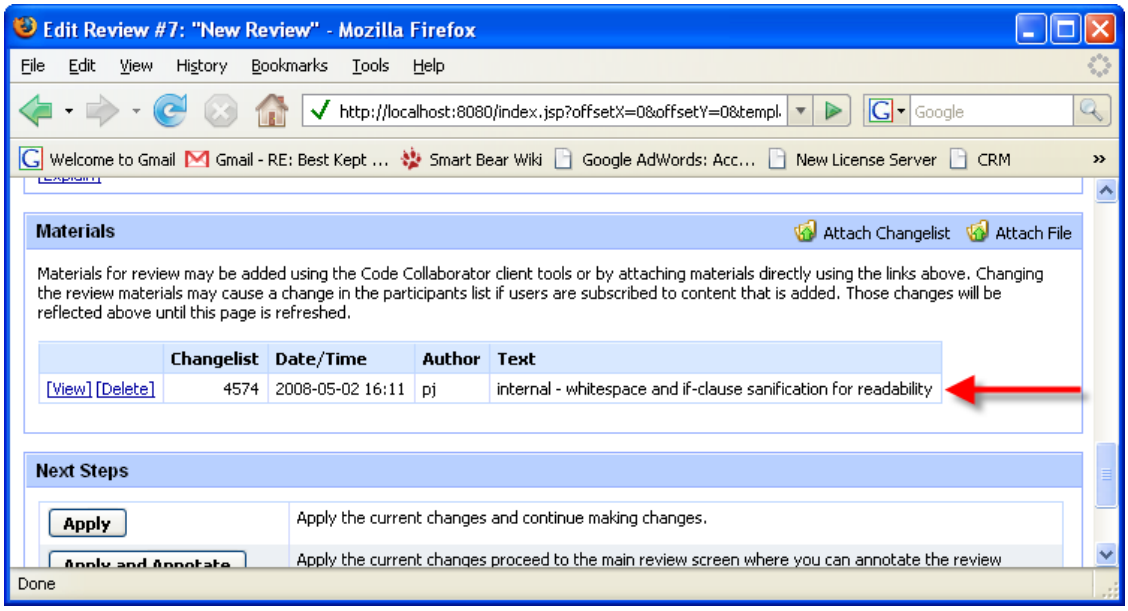
- Local file changes not yet checked into version control
- Changes already checked into version control
- Differences between files on your disk
- Files from your local disk
- Review checklists and test scripts
- Requirements and architecture specifications
- URLs
- Images
- PDFs

In order to attach materials, you must firstApply 146 changes to a review before you attempt to attach your materials.

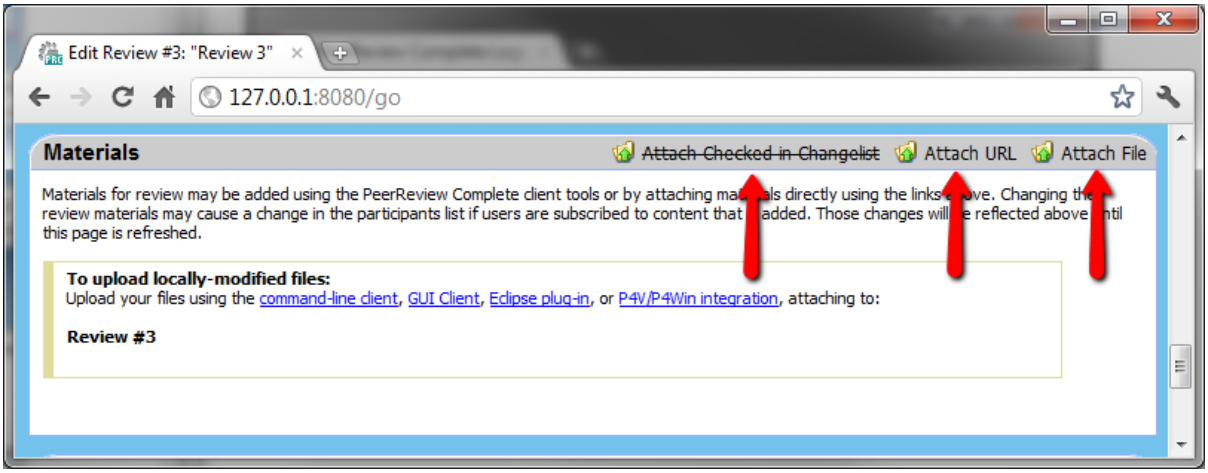
The most common way to get files attached to a review is to use one of the client components: Command Line Client 196, GUI Client 189, or Perforce Plug-in 366. All of these clients can integrate with version control and/or easily upload local files or file-differences to the server.

© 2003-2011 SmartBear Software

Local changes from files controlled by Subversion have been uploaded into this review via client.


Edit Review #7: "New Review" - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:8080/index.jsp?offsetX=0&offsetY=0&templ
Google
Welcome to Gmail
Gmail - RE: Best Kept ...
Smart Bear Wiki
Google AdWords: Acc...
New License Server
CRM
Materials
Attach Changelist
Attach File
Materials for review may be added using the Code Collaborator client tools or by attaching materials directly using the links above. Changing the review materials may cause a change in the participants list if users are subscribed to content that is added. Those changes will be reflected above until this page is refreshed.
Changelist
Date/Time
Author
Text
[View] [Delete]
4574
2008-05-02 16:11
pj
internal - whitespace and if-clause sanification for readability
Next Steps
Apply
Apply the current changes and continue making changes.
Apply the current changes proceed to the main review screen where you can annotate the review
Done

It's also possible to upload content from your web browser. You can attach a changelist (if your Server-side Version Control System has been configured), a URL, or a file. Just click the appropriate button indicated by the red arrows below:


Edit Review #3: "Review 3"
127.0.0.1:8080/go
Materials
Attach Checked-in Changelist
Attach URL
Attach File
Materials for review may be added using the PeerReview Complete client tools or by attaching
this page is refreshed.
To upload locally-modified files:
Upload your files using the command-line client, GUI Client, Eclipse plug-in, or P4V/P4Win integration, attaching to:
Review #3

When you use "Attach Checked-in Changelist," a dialog box prompts you to select a pre-configured version control server [101] and the unique changelist ID you want to attach:

© 2003-2011 SmartBear Software

If there's an error retrieving that version or communicating with the server, you'll be presented with an error message. You can attach any number of changelists to a single review, and you can attach changelists from different version control servers to a single review.

The result is exactly the same as if you implemented this operation from a client-side tool, for example using the command-line `ccollab addchangelist` 218. The advantage of performing this action from a web browser is that you don't need to install and configure the client.

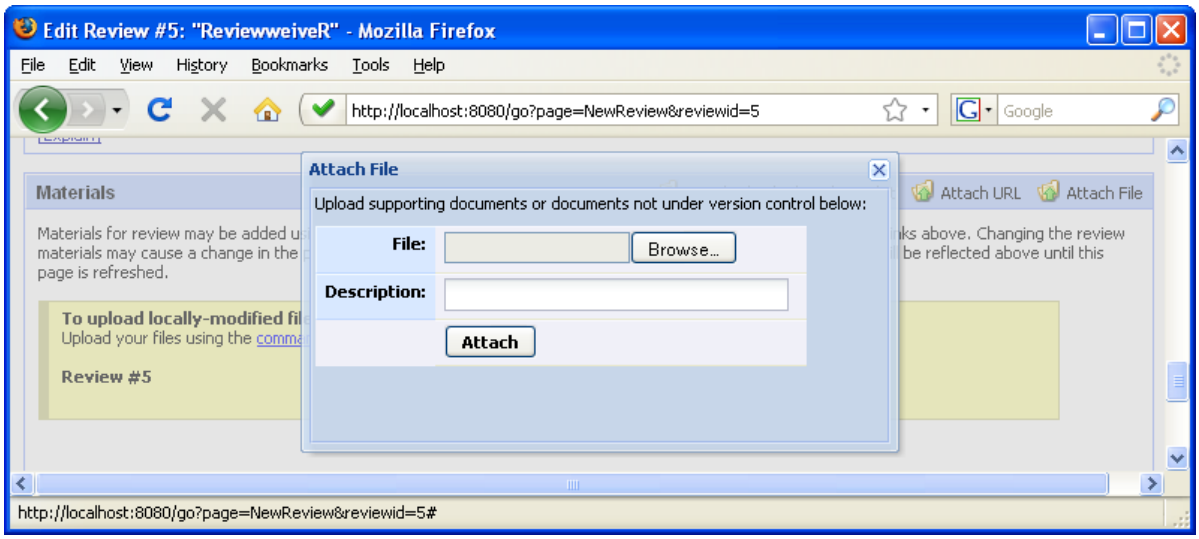
You can attach live URL links using "Attach URL":

Instead of pulling a copy of the content, attaching a live url will display the live contents of the link during the review. This feature is useful for accessing a document on a Wiki or document management system, particularly if you link to a specific version of that document.

The URL must be a valid web URL, which should include the protocol specification (either http:// or https://). The Attach a live URL dialog does not confirm that the resource named is actually available. In fact, since it's a live link, the contents and availability of the document can change at any time. An error is displayed only if the URL given is malformed.

You can also attach a file (images, PDFs, and binary file types) from your local hard drive using the "Attach File" link:

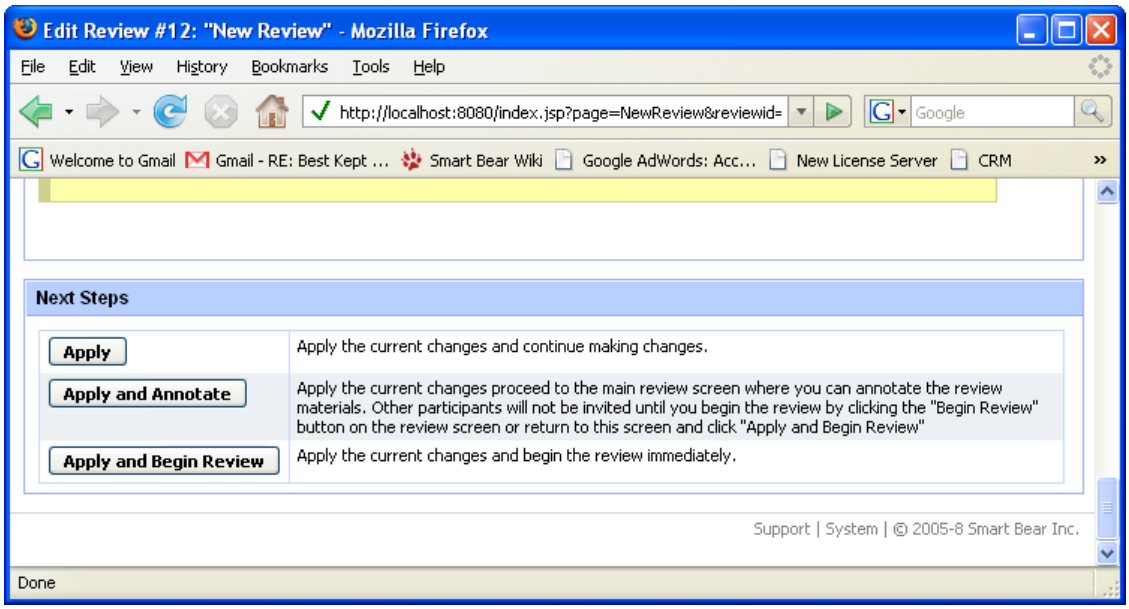
© 2003-2011 SmartBear Software



Once you have attached your document, the attached documents will be listed.

Completing the Review

The last step allows you to start the review:


Edit Review #12: "New Review" - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:8080/index.jsp?page=NewReview&reviewid=
Google
Welcome to Gmail
Gmail - RE: Best Kept ...
Smart Bear Wiki
Google AdWords: Acc...
New License Server
CRM
Next Steps
Apply
Apply the current changes and continue making changes.
Apply and Annotate
Apply the current changes proceed to the main review screen where you can annotate the review materials. Other participants will not be invited until you begin the review by clicking the "Begin Review" button on the review screen or return to this screen and click "Apply and Begin Review"
Apply and Begin Review
Apply the current changes and begin the review immediately.
Support | System | © 2005-8 Smart Bear Inc.
Done

You will have three buttons to click.

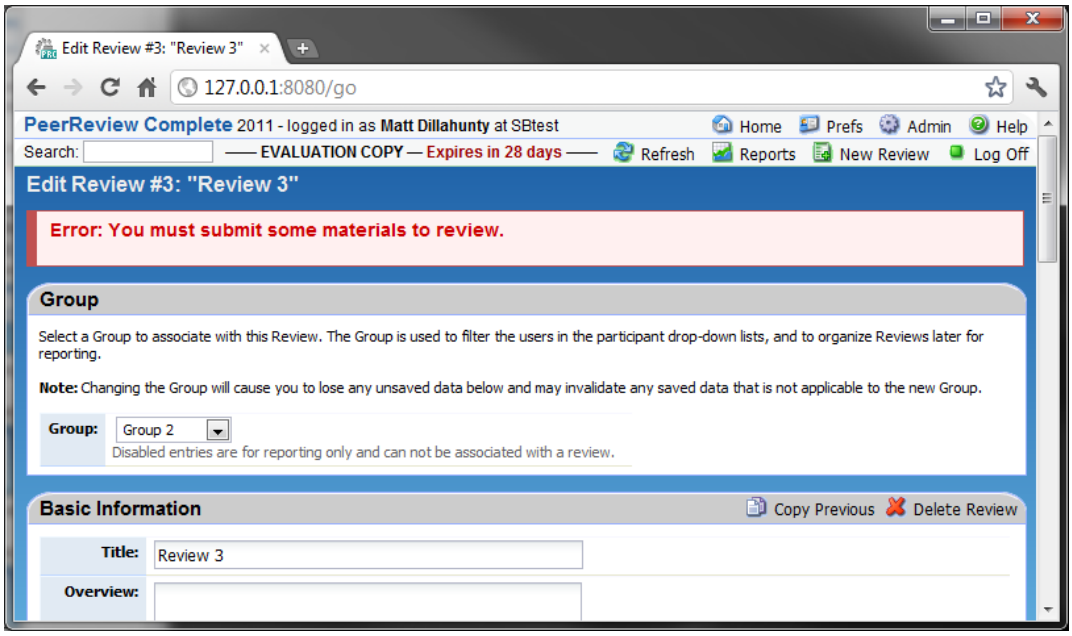
By clicking "Apply," you will apply the current changes that you have made, while remaining on the current page, allowing you to continue making changes.

© 2003-2011 SmartBear Software

By clicking "Apply and Annotate," you will apply the current changes, and the browser will take you to the main review screen so that you may annotate the review materials. However, you will be the only person viewing and editing the review until you click "Begin Review." This means you can go around to each of the files and leave comments for other users to see. This has proven to be an effective way of finding your own defects, thereby making the actual review more efficient for the other participants.

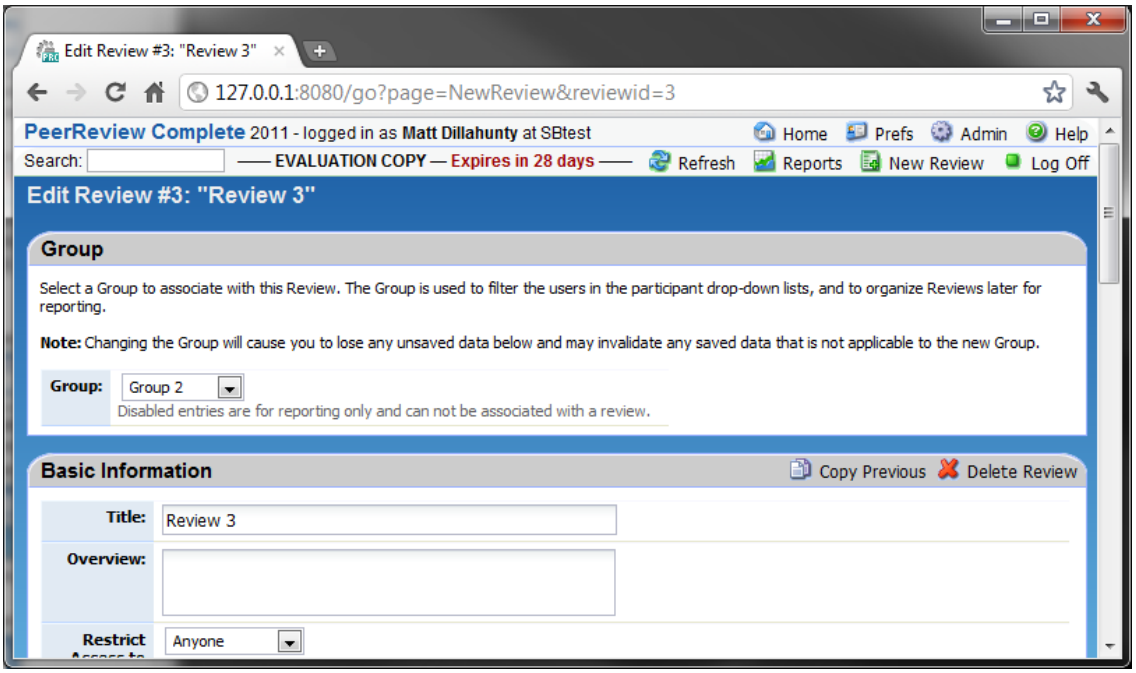
Click "Apply and Begin Review" to begin the review. Until this point, no one else has seen the review in their Action Items[139], and no notifications have been sent out. As soon as you click the button, the review will start.

If you go to this step without having properly completed the steps before, you will see explanatory error messages in red:


Edit Review #3: "Review 3"
127.0.0.1:8080/go
PeerReview Complete 2011 - logged in as Matt Dillahunty at SBtest
Home
Prefs
Admin
Help
Search:
EVALUATION COPY — Expires in 28 days
Refresh
Reports
New Review
Log Off
Edit Review #3: "Review 3"
Error: You must submit some materials to review.
Group
Select a Group to associate with this Review. The Group is used to filter the users in the participant drop-down lists, and to organize Reviews later for reporting.
Note: Changing the Group will cause you to lose any unsaved data below and may invalidate any saved data that is not applicable to the new Group.
Group:
Group 2
Disabled entries are for reporting only and can not be associated with a review.
Basic Information
Copy Previous
Delete Review
Title:
Review 3
Overview:

© 2003-2011 SmartBear Software

Canceling a review before it starts
To cancel a review before it begins, use the Delete Review button on the top right of the "Basic Information" box:


Edit Review #3: "Review 3"
127.0.0.1:8080/go?page=NewReview&reviewid=3
PeerReview Complete 2011 - logged in as Matt Dillahunty at SBtest
Home
Prefs
Admin
Help
Search:
EVALUATION COPY — Expires in 28 days
Refresh
Reports
New Review
Log Off
Edit Review #3: "Review 3"
Group
Select a Group to associate with this Review. The Group is used to filter the users in the participant drop-down lists, and to organize Reviews later for reporting.
Note: Changing the Group will cause you to lose any unsaved data below and may invalidate any saved data that is not applicable to the new Group.
Group:
Group 2
Disabled entries are for reporting only and can not be associated with a review.
Basic Information
Copy Previous
Delete Review
Title:
Review 3
Overview:
Restrict
Anyone

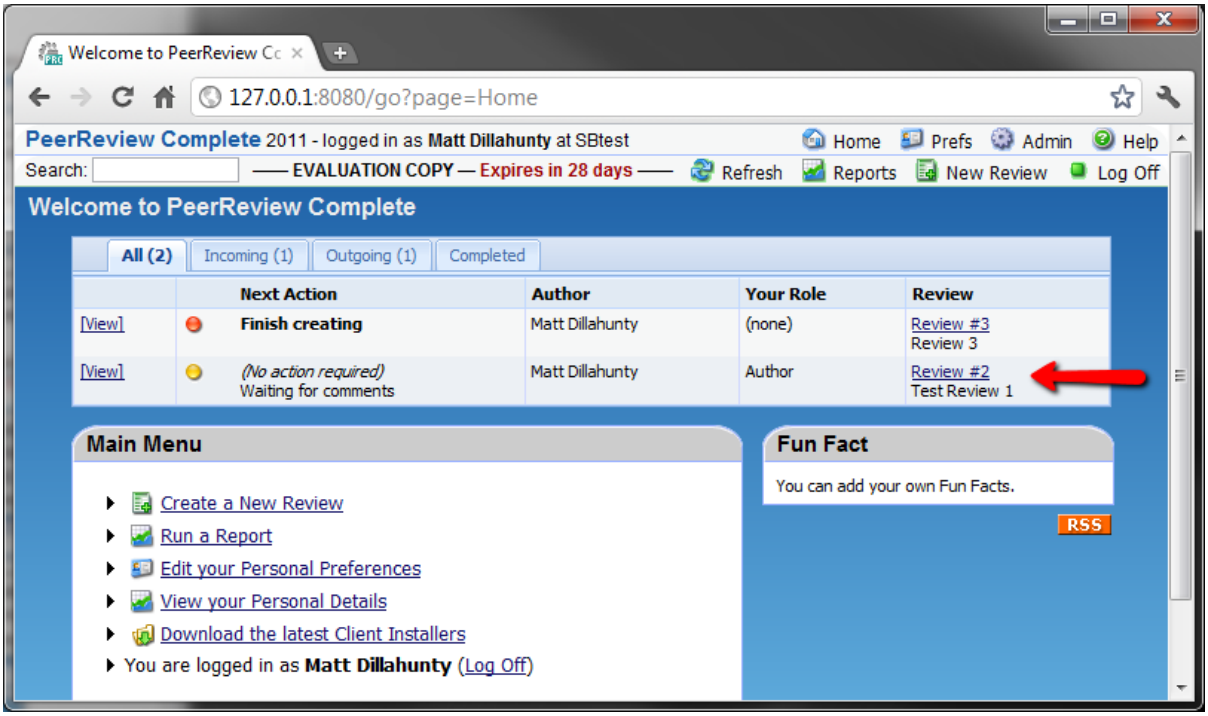
3.2.4 Review Summary Screen

The Review Summary Screen is presented any time you're viewing a review that's not still in the planning stage. It is split up into several sections:

- General Information 149
- Participants 150
- Defect Log 151
- Overall Chat 152
- Review Materials 152
- Moving On 156

© 2003-2011 SmartBear Software

You will know that you're part of a review because it will show up in your Action Items list 139:


Welcome to PeerReview Cc
127.0.0.1:8080/go?page=Home
PeerReview Complete 2011 - logged in as Matt Dillahunty at SBtest
Home Prefs Admin Help
Search:
EVALUATION COPY — Expires in 28 days
Refresh Reports New Review Log Off
Welcome to PeerReview Complete
All (2) Incoming (1) Outgoing (1) Completed
Next Action Author Your Role Review
[View] Finish creating Matt Dillahunty (none) Review #3 Review 3
[View] (No action required) Waiting for comments Matt Dillahunty Author Review #2 Test Review 1
Main Menu
Create a New Review
Run a Report
Edit your Personal Preferences
View your Personal Details
Download the latest Client Installers
You are logged in as Matt Dillahunty (Log Off)
Fun Fact
You can add your own Fun Facts.
RSS

General Information Section

The General Information section shows the overall review information. The image below shows the general information section for a completed review:

Most of the fields mirror the title and custom fields supplied by the review creator.

You might be able to **Edit** the title and custom fields by clicking the "Edit" button on the top right. To have permission to do this, you must be either a participant in the review or an administrator, and the review must be in progress (i.e. not canceled or complete), and the role rules 107 for your role must allow you to do this. In this case, the `Edit` toolbar button lets you edit the fields. Otherwise, the button will be drawn disabled as shown above, and if you click it, an error message will explain why you can't edit the fields.

Clicking on the **Details** link will direct to the review's Review Details Report 179 tailored to the open review.

© 2003-2011 SmartBear Software

You might be able to **Cancel** the review. To have permission to do this, you must be authorized 79 to do so by the administrator, and the review must be in progress (i.e. not already canceled or complete). In this case, the `Cancel Review` button is drawn disabled, and if you click it, an error message will explain why you can't edit the fields.

> **Warning:** Once a review is canceled it is canceled forever. You will have to create a brand new review. Users will still be able to find this review when searching and can view it in a read-only capacity, but the review cannot be restarted.

You might also be able to **Re-open** a review after a review is finished. You must be allowed 79 to do so by the system administrator. In this example above, the user has the ability to reopen the review. However, if it is not allowed, the button will be drawn disabled.

Participants Section

The Participants section lists all participants in the review, their status, and their role:

Each user is listed according to his role in the review. User names are drawn ~~crossed out~~ if that user has already indicated that they are finished with the review. Users with configured email addresses will be linked to their email address, and user's phone numbers are displayed in tool-tips.

Also given in parenthesis are user initials. Initials are used throughout the review interface to identify users without taking up the screen space to print out entire names. Initials are determined automatically for each review so that no two users conflict. In the example above, "Isaac Bon" and "Ivan Brock" share the same initials, so their indicating initials are `IsB` and `IvB`. If no two users share the same initials, the standard two character initials will be given.

You can also use the **Poke** feature to send someone a notification immediately. To use this feature, click the `[Poke]` link to the right of the participant you wish to poke. This may be useful when a certain participant has stalled the review or when you would like to invite a "finished"/"waiting" participant back into the review.

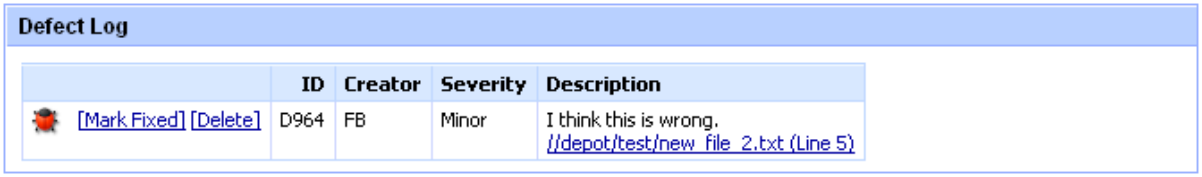
You might be able to **Edit** the participant list. To have permission to do this, you must be either a participant in the review or an administrator, and the review must be in progress (i.e. not canceled or complete), and the role rules 107 for your role must allow you to do this. If you're allowed, the `Edit` toolbar button lets you edit the list. Otherwise, the button will be drawn disabled, and if you click it, an error message will explain why you can't edit the fields.

When you change the list, any users removed will receive a notification that they've been removed and any users added will receive a notification that they've been added. You can add or remove yourself from a review, and indeed this is the most common way to "hand off" a review to someone else.

© 2003-2011 SmartBear Software

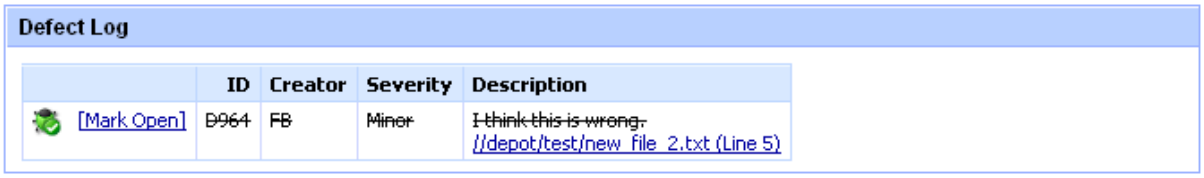
Defect Log Section

The Defect Log lists all defects found in the review:



Initially the log will be empty. As defects are created for the review as a whole or on individual files and line numbers, all defects are collected and listed here. For defects associated with particular files, links to that file and line number appears in the table for fast access. All defect custom fields are shown in the table.

When, as the reviewer, you've verified that a defect you found has been fixed, you can indicate this by clicking `[Mark Fixed]`. You don't want to *delete* the defect in this case -- it was still a defect -- but you need to indicate that "everything is fine now." The defect is then redrawn to reflect this:



You can mark the defect open again if you change your mind. Open the defect by clicking`[Mark Open]`.

You can also externalize 171 a defect if you want to indicate that the defect will be fixed sometime after the review is complete. In this case, it will be indicated by a green bug with a blue arrow icon as shown below:

Only certain participants will be able to mark defects fixed. Which roles are allowed to do this depends on the system configuration 107. Also the user that created the defect is allowed to mark it fixed, and administrators are always allowed to mark any defect fixed.

Most of the time you will create defects 170 and mark them fixed 171 on individual lines.

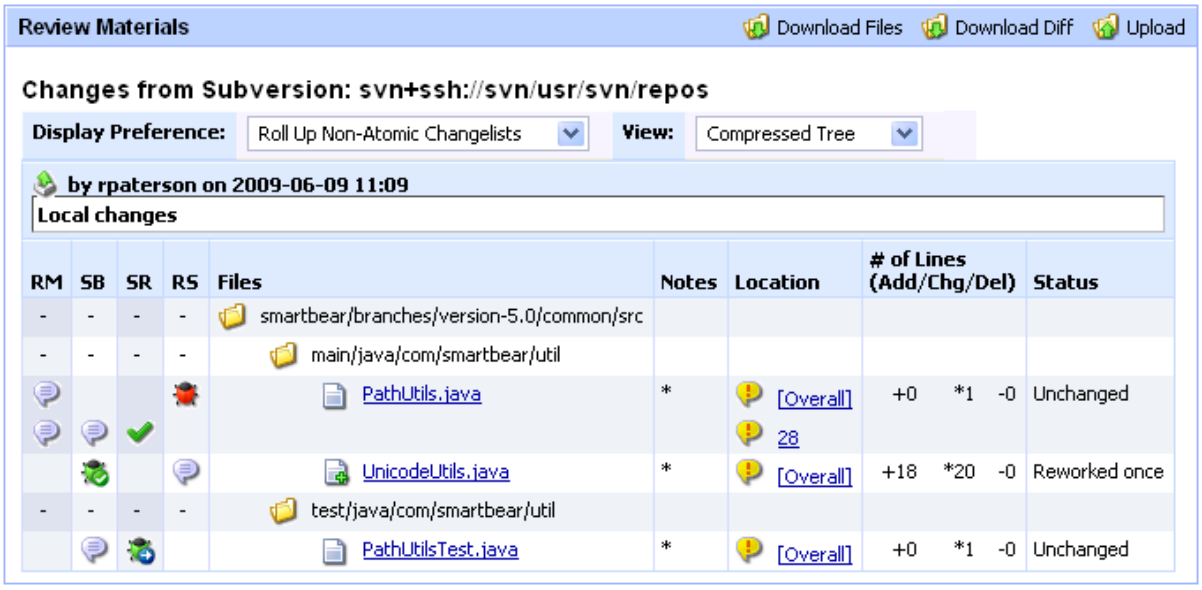
© 2003-2011 SmartBear Software

Overall Review Chat

You can chat with other users about this review as a whole, rather than chatting on individual lines:

There's much to say about chat and defects, but this interface is identical to the one in the Review Chats 167, so please see that section for details.

Review Materials Section



The Review Materials section is the heart of the review: All files are displayed and users can view content and differences and chat and create defects on specific lines of a file.

Toolbar

The toolbar has 3 buttons:

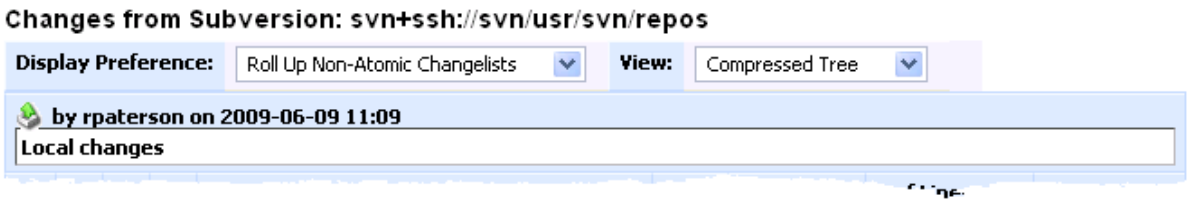
© 2003-2011 SmartBear Software

- Download Files - This downloads a ZIP file to the local hard drive containing all files *with subdirectory structure preserved*. This means you can **test** the proposed file changes locally: Just download the ZIP file and expand it in your own development environment. This is also useful in the **single-committer** model for when you want to actually commit the changes. This can take the place of a patch file.

> **Warning:** Downloading files to your local hard drive can have unintended consequences. Make sure you don't have changes of your own that you're overwriting. Also, remember that the author of these changes might not be synched to exactly the same versions of all files in version control, so if your local test fails this might be the reason.

You can also download individual file versions from the diff viewer 157, but the Download Files toolbar link is the more common way because you get all files at once.

- Download Diff - This downloads a unified diff of all files in the review.
- Upload Files - You might be able to **Upload** additional files to the review. You can do this any time using one of the various client components, or you can use the toolbar button to be prompted further or to upload a file from your local hard drive. This button might be disabled if the review is complete or if you are not a participant; click the disabled toolbar icon to get an error message explaining exactly why you cannot upload files.



Section

All of the materials are displayed separately in subsections: Version control uploads, individual file uploads, etc. In each case, you can see any applicable version control information (author, date, comments, changelist ID) and all the files.

When multiple changes have been uploaded, you can set the *Display Preference* to control how they are displayed:

- Do Not Roll Up
  All uploaded changes are displayed. Files are grouped with other files that were uploaded at the same time. In the case of version control systems with atomic changelists, this means files are grouped by changelist. For other version control systems, each uploaded file-list appears separately. This selection is the default behavior for SCM systems with atomic changelists unless the review creator has specifically requested the rolled up view.

  Default Roll Up settings are SCM specific. Atomic changelists are not rolled up by default, and non-atomic changelists are rolled up by default. To show hidden changelists, click the "Show previously-uploaded changelists" link. To hide them again, click the "Hide previously-uploaded changelists" link.

© 2003-2011 SmartBear Software

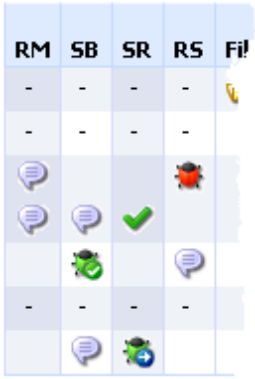
- Roll Up All Changelists
  This is available when the version control system doesn't have the concept of an "atomic changelist" before files are checked in (e.g. Subversion, ClearCase, CVS). In this option, all uploads are "rolled up" into a single list of files. A "status" column lets you know the status of each file. For example, a file might have been uploaded originally but wasn't changed during the last upload; in this case the status would be "unchanged." If a file was changed since the last upload the status would be "reworked."

  This is the default selection for SCM systems that have no atomic changelists and can also be made the default for a specific review by the review creator.
- Roll Up Non-Atomic Changelists
  This is available with version control systems that have atomic changelists after check-in (e.g. Subversion). In this option, atomic changelists (after check-in) are not rolled up but non-atomic changelists (before check-in) are rolled up. The "status" column lets you know the status of each file as described above.

  This is the default selection for SCM systems have atomic changelists only after check-in.

The *View* controls how the file paths are displayed. Note if the files are under version control, the version control server paths are listed rather than local hard drive paths -- presumably the version control server paths will make more sense to other users. The default *View* is controlled by a user-level preference.

- Compressed Tree - Displays the directory structure of the files while saving as much space as possible. Files are always displayed on a new line. Parent directories are "compressed" where possible to save space.
- Tree - Classic "Tree" view typically used for navigating file systems.
- Flat - Each row contains the entire file path, with only the topmost common path pulled out.



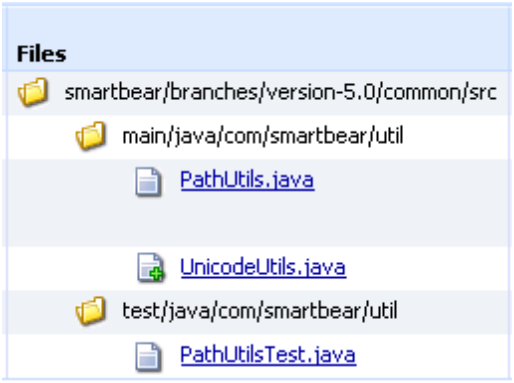
Conversation status icons

Columns next to each file and line-number conversation indicate which users have commented and what their current status is. User names are abbreviated to take up less space.

Icon	Meaning
	User has clicked "Accepted"
	User has made a Comment

© 2003-2011 SmartBear Software

	User has created a Defect, which is currently "Open"
	User has created a Defect, which is currently marked "Fixed"
	User has created a Defect, which is currently marked "Tracked Externally"



File list

The file list shows the files in this changelist. This part of the screen may look different depending on the selected *View* 154. The icon next to the file name indicates what operation was performed to the file.

Icon	Meaning
	File was modified
	File was added
	File was deleted
	File was reverted

Click on a file to open the content 157 of that file and view that all comments.

Notes	Location	# of Lines (Add/Chg/Del)	Status
*	[Overall] 28	+0 *1 -0	Unchanged
*	[Overall]	+18 *20 -0	Reworked once
*	[Overall]	+0 *1 -0	Unchanged

Notes, Location, Metrics, and Status

The *Notes* column allows any user to annotate the files. To edit the note click in the note column. This is typically used for messages to all reviewers such as "review this file first", or "ignore this file".

© 2003-2011 SmartBear Software

The *Location* column indicates where within the file the conversation represented by this row is located. Yellow alert bubbles mean there is activity on that line that the current user has not yet marked read. Click on a line number to open the content [157] of that file and view that conversation.

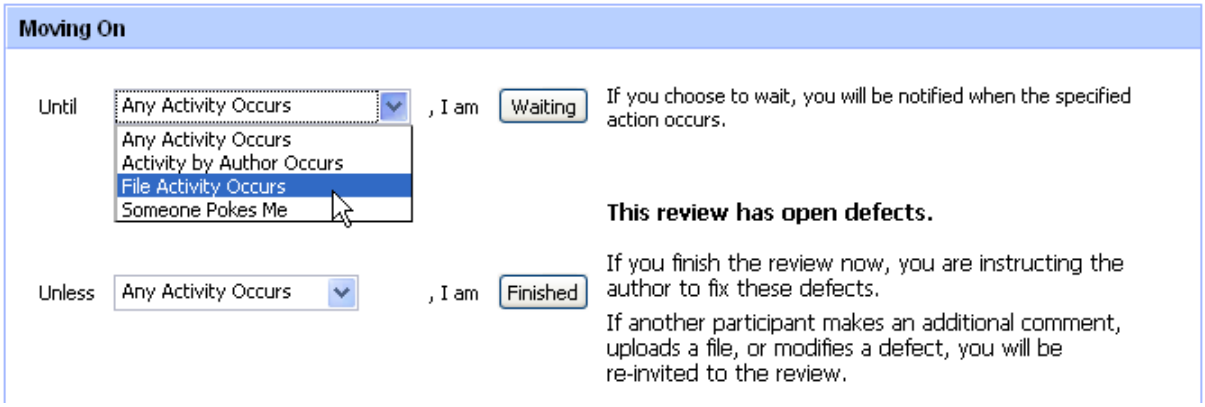
The column labeled *# of lines (Add/Chg/Del)* shows the metrics of how many lines have been added, changed, and deleted since the previous version in version control, with special cases for situations like added or deleted files.

The S*tatus* column shows cumulative status of a file based on combining the actions of all previous changelists uploaded to the review.

Moving On

When are you finished with a review? When does a review move on to another phase? At the bottom of the Overview screen, you'll find a "Moving On" section that allows each review participant to specify whether he/she is **Waiting** on review activity or has **Finished** with the review (with the option to re-engage based on future activities). These choices move reviews into different phases. The options in "Moving On" vary depending on the phase of the review and the role of the participant. "Moving On" also allows you to control - to a certain extent - the notifications you receive for a particular review.

The example review [136] gives a high-level description of the phases of a review and how you get from one phase to another.


Moving On
Until
Any Activity Occurs
, I am
Waiting
If you choose to wait, you will be notified when the specified action occurs.
Any Activity Occurs
Activity by Author Occurs
File Activity Occurs
Someone Pokes Me
This review has open defects.
Unless
Any Activity Occurs
, I am
Finished
If you finish the review now, you are instructing the author to fix these defects.
If another participant makes an additional comment, uploads a file, or modifies a defect, you will be re-invited to the review.

You can choose to Wait, in which case PeerReview Complete notifies you when the specified action occurs, or you can choose Finished to finish your part in the review unless one of the specified actions occurs, in which case PeerReview Complete will send a notification to re-engage you. The action options are:

- **Any Activity Occurs** - any action that occurs in the review; for example, comments, file uploads, or added participants
- **Activity by Author Occurs** - any action that occurs in the review that is initiated by the author
- **File Activity Occurs** - any action that occurs to change the file content of the review; for example, the upload of a new file or the upload of a new version of an existing file
- **Someone Pokes Me** - only notifies you in the event that you are poked

© 2003-2011 SmartBear Software

3.2.5 Reviewing Materials

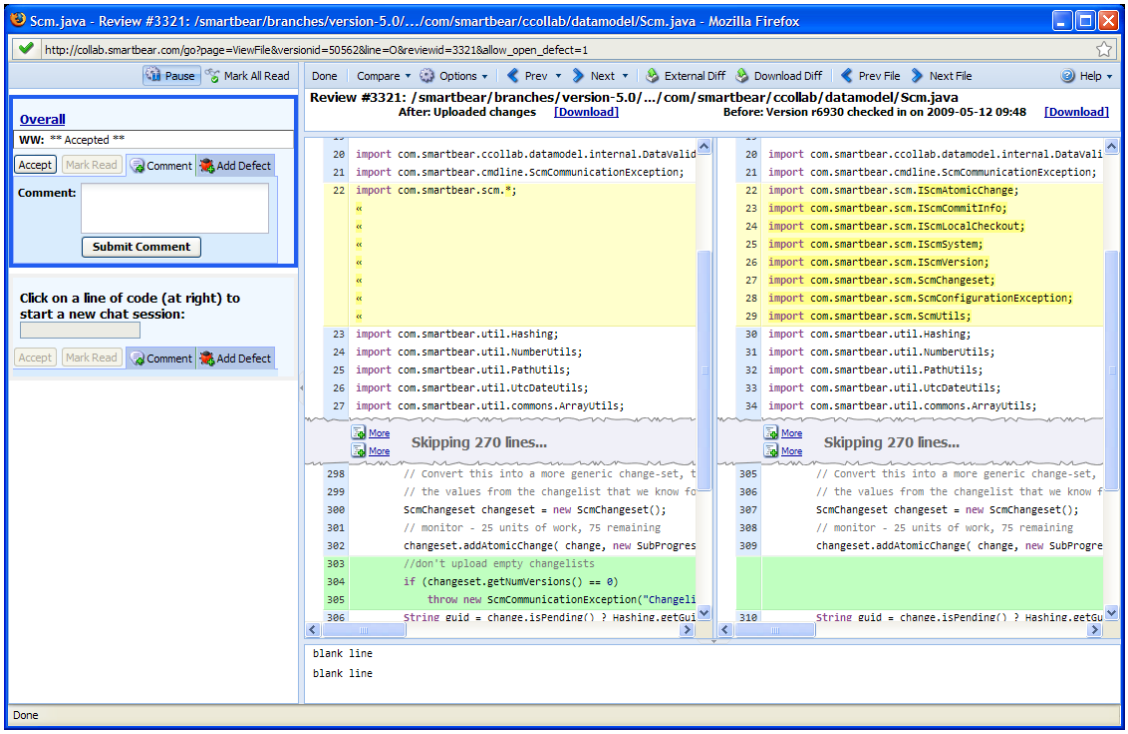
The Review Materials screen, or the diff viewer, is where most of the reviewing takes place. The commenting format and functionality will differ depending on what file type you are reviewing. We support the following:

- Text Files 157 (any)
- Images 162 (JPEG, PNG, and GIF)
- Documents 164 (PDF, and any document type under Windows)
- URLs 166 (http and https)

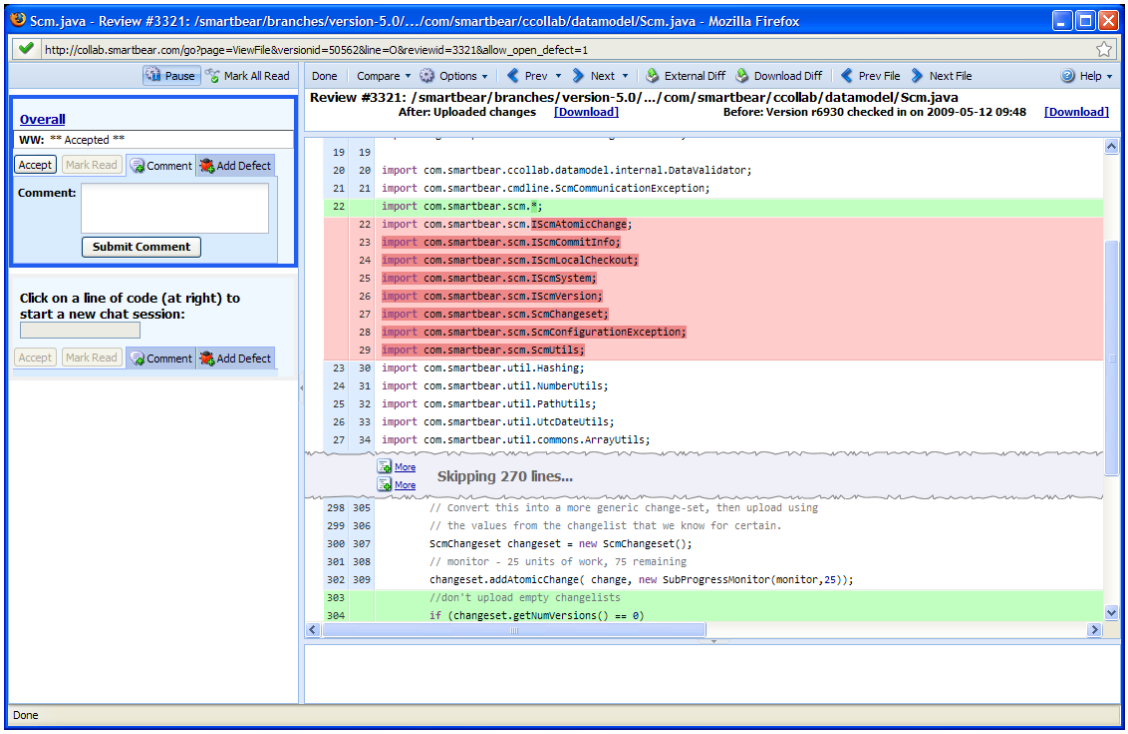
3.2.5.1 Reviewing Text Files

When reviewing text files, the review screen becomes the diff viewer. Here, you can see file content, differences with previous versions of the file, and all comments and defects for the file all on one screen.

The PeerReview Complete diff view displays any text either side-by-side or in an over-under orientation. Both views have all the same options and functionality, so users should feel free to use whatever is most comfortable. Since differences in the content could affect users' orientation preference, changing the orientation is as simple. Just select Options > Diff View in the top toolbar.


Scm.java - Review #3321: /smartbear/branches/version-5.0/.../com/smartbear/ccollab/datamodel/Scm.java - Mozilla Firefox
Review #3321: /smartbear/branches/version-5.0/.../com/smartbear/ccollab/datamodel/Scm.java
Skipping 270 lines...
Skipping 270 lines...

Standard View with Side-by-Side Orientation



Standard View with Over/Under Orientation

The over and under orientation merges both versions of the file together. Unchanged lines are displayed as one, while added and deleted lines are highlighted with the appropriate line number for the version. The left two columns in blue correlate to the versions listed above the main file view respectively.

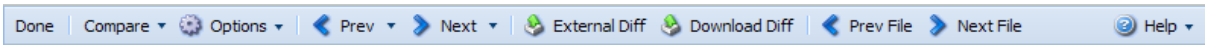
Chat Session

On the left of the main diff viewer page, there is a section for chat threads, where you can view and make comments. This will be explained in further detail in thenext section 167. If you would like to temporarily hide this section to leave more screen space to look at the uploaded file, you do so by double clicking the separator between the chat and the content or by clicking on the arrow in the separator as indicated below:

© 2003-2011 SmartBear Software

Menubar

Right at the top, above the main file view, a menubar is displayed with a few possible actions. Buttons with blue triangles to the right will display drop-down menus when clicked on.

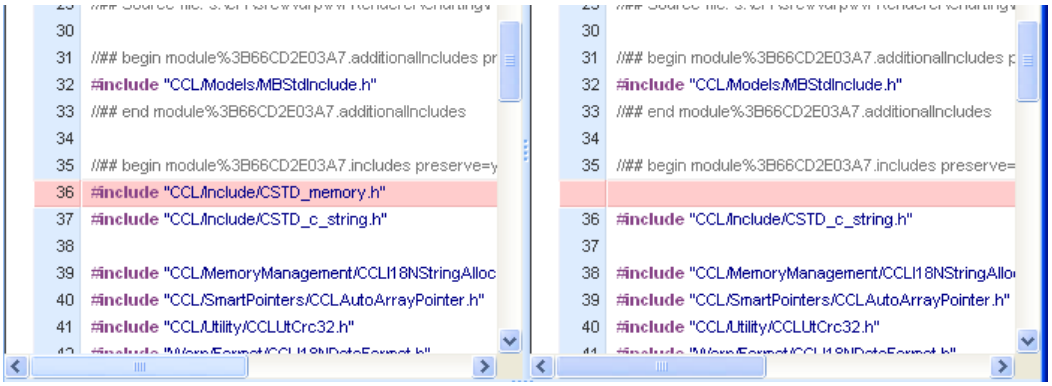


- **Done** - Clicking this button closes the diff viewer.
- **Compare** - This menu will allow you to choose which file content are shown in the main file view. You can choose any two versions of the file in the review to compare, or set both left and right sides to the same version to view it alone.
- **Options** - These options refer to the diff viewer preferences that can also be found with detailed descriptions in the User Preferences section 131. These options are also available in the classic view above the main file viewer.
- **Prev/Next** - The Previous and Next buttons allow you to quickly jump to a previous or the next difference, defect, or comment depending on which one(s) is(are) checked. This is especially handy when "Skip Unchanged Lines" is disabled and you want to leap to the next diff without having to manually scroll through the file looking for it.
- **External Diff** - This opens the diff in an external diff viewer. This feature requires the External Diff Viewer launcher client 280.
- **Download Diff** - This downloads the changes as a Unified diff.
- **Prev File/Next File** - New in PeerReview Complete 5.0, this allows you to switch between the files in the review *without* closing the diff view and finding the next file in the Review Summary 148 screen.
- **Help** - Here, you can find helpful legends 160, keyboard shortcuts 160, and a quick link to the owner's manual.

© 2003-2011 SmartBear Software

The File Content View

To the right of the chat session, there is the main file view. This is where you can view file content and differences. The format of the main file on your local PeerReview Complete screen may be different depending on how your user preferences 131 are configured.

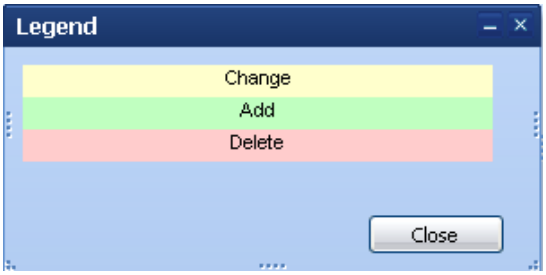


If differences are being shown, the interface automatically scrolls to the first difference (unless you got to this screen by clicking on a particular line number, in which case the screen will be centered on that line number).

- **Over-under Comparison** - Below the file content is a small area which shows the currently selected line in over-under view. This is most useful when the content view is in side-by-side mode, but the differences are not immediately clear in that view.

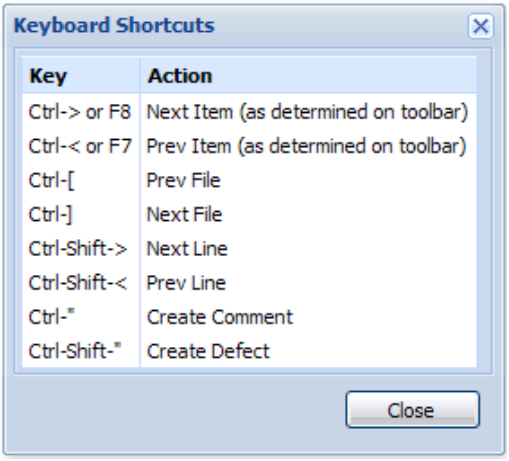
Legends

The differences are noticeably highlighted and color coded as explained by the legend. To view the legend, click on "Help" then "Legend," and a pop-up box should appear.



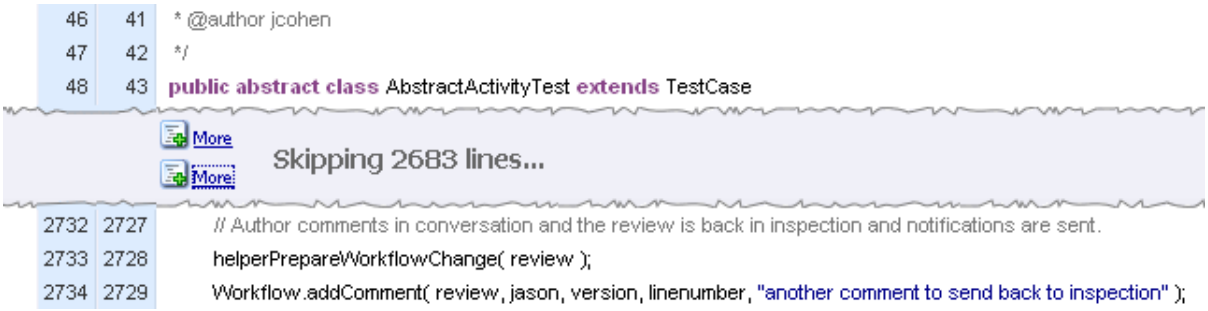
The diff view is completely navigable from the keyboard. You can move around the file, jump to changes, and make comments and defects. To get the list of keyboard shortcuts, click the "Help" link in the menu bar, then "Keyboard Shortcuts."

© 2003-2011 SmartBear Software



Skip Unchanged

If the "skip unchanged 131" option is enabled, unchanged lines will be hidden in the diff viewer as shown below:


46 41 * @author jcohen
47 42 */
48 43 public abstract class AbstractActivityTest extends TestCase
More
More
Skipping 2683 lines...
2732 2727 // Author comments in conversation and the review is back in inspection and notifications are sent.
2733 2728 helperPrepareWorkflowChange(review);
2734 2729 Workflow.addComment(review, jason, version, linenumber, "another comment to send back to inspection");

If you'd like to get more lines of context, just click either "More" link. The top link will show the next 25 lines of context, while the bottom link will show the last 25 lines of context.

Syntax Coloring

With Syntax Highlighting chosen 132, some file types are detected automatically and are marked up with syntax coloring. If the file type isn't recognized, it's simply displayed without highlighting.

The file viewer will color language syntax for certain file types. Although PeerReview Complete works with any text files, this feature helps read these languages and formats more easily:

- ASPX
- C
- C++
- CSS
- HTML
- Java
- JavaScript
- JSP
- Perl

© 2003-2011 SmartBear Software

- PHP
- Python
- Ruby
- SGML
- Shell
- Visual Basic (VB)
- XML

3.2.5.2 Reviewing Images

Images can be reviewed within the tool and not as external binary files. Image files must be renderable by the browser and be configured by the administrator 83 to be treated as images. By default, JPEG (.jpg and .jpeg), GIF (.gif), and PNG (.png) files are treated as images.

Uploading Images
To review images, just attach 145 them to the review as you would any other file, and when you open the diff viewer, the content area will display the images for review.

Making Comments
To comment on the images, simply click on the image at a point where you would like your comment to appear and begin typing your comment. A blue pushpin ( ) will appear on the image to indicate the location of the currently selected conversation. Other pushpins indicate other conversations on the image:

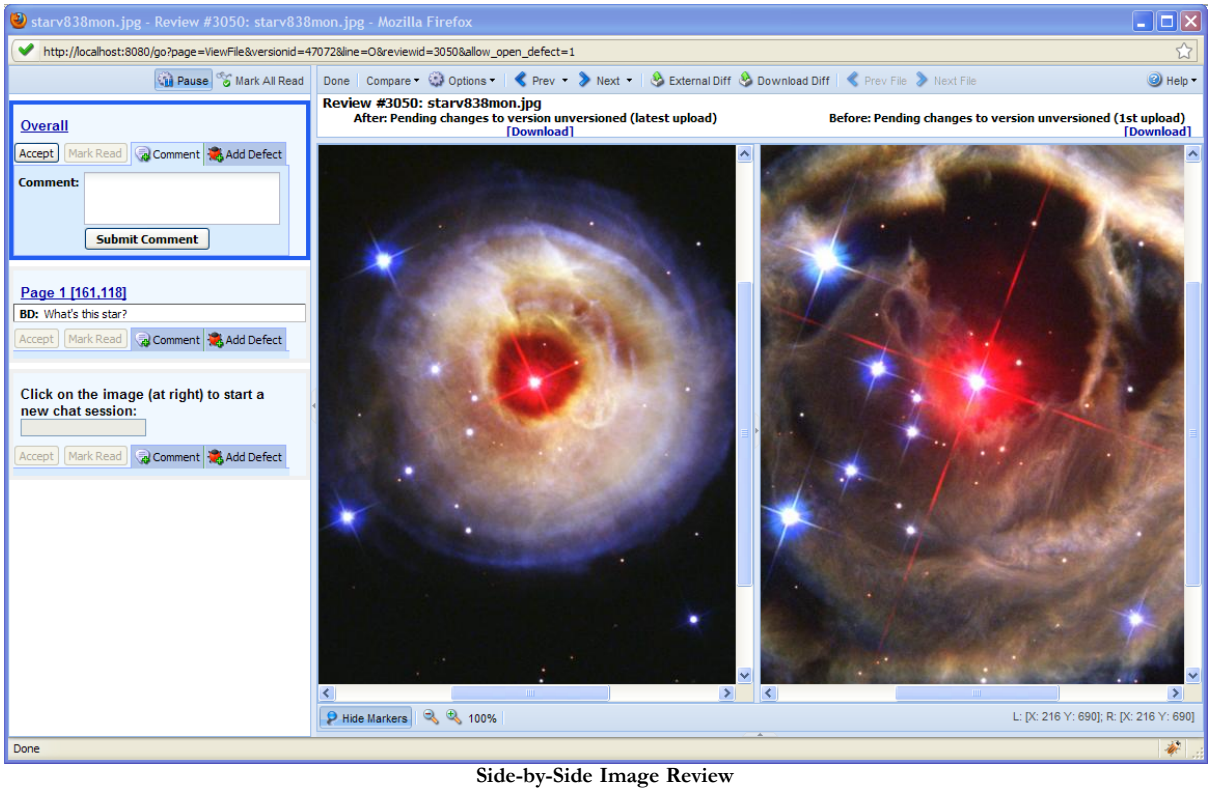
Red	Open Defects
Yellow	Unread Comment
Green	Fixed Defects or External Defects
Grey	No Defects

The pins may be turned off by clicking the pushpin toggle in the image review toolbar at the bottom of the content pane. This is useful when the pushpins cover important parts of the image.

Other than the difference in location markers, review chats 167 are handled in the same manner as in reviewing text files.

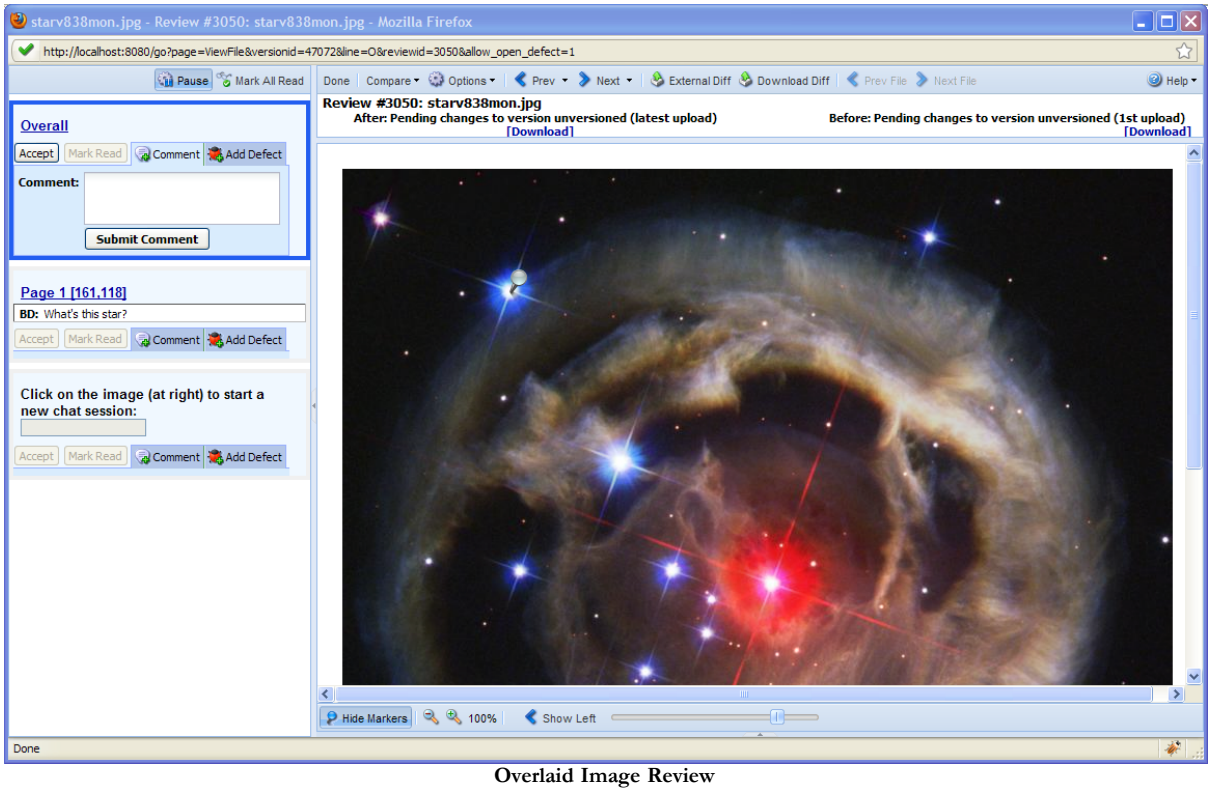
Comparison
Images can be reviewed in two different manners: **side-by-side** or **overlaid**. Toggling between the two layouts is as simple as changing the Diff View setting in the Options menu.

© 2003-2011 SmartBear Software



Side-by-Side Image Review

The **side-by-side view** is useful for viewing the two images, before and after, as completely separate images and when you need to look at both images at the same time.



Overlaid Image Review

© 2003-2011 SmartBear Software

The **overlaid view** is useful for seeing the subtle differences between two images, including registration (alignment) differences and color variation. The **slider** on the image review toolbar allows you to adjust to opacity of the top image, thereby allowing you to blend the two images together. To quickly toggle between the two images, toggle the "**Show Left/Show Right**" button on the toolbar.

Zoom
Both the side-by-side and overlaid view support zooming the images in the display. The image review toolbar includes two magnifying glass buttons for zoom and a label displaying the current zoom scale. Zoom out is useful if the image is too large to display in the browser window without scrolling. Zoom in is useful for more precise placement of the comment markers. Whether zoomed in or out, the comment markers still refer to pixel locations in the image at 100% zoom, so subpixel comments are not supported. If a zoom operation would cause the currently selected location to scroll offscreen, the image will be scrolled to recenter the selected location.

The image rescaling for the zoom function is provided by the browser and may introduce artifacts depending on the type of image and the scaling algorithm chosen by the browser.

Note: The images "reviewed" in the above screenshots were taken from the Hubble Space Telescope and are used courtesy of NASA and STScI. VisitHubbleSite for more great images from Hubble.

3.2.5.3 Reviewing Documents

PeerReview Complete has the ability to review PDFs. Additionally, PeerReview Complete v6.0 allows Windows users to add any document type, such as Microsoft Office documents, to a review usingPrint To Review 282. Other document types on other platforms can be reviewed by converting them to PDF using a conversion tool. Popular converters include (but are not limited to):

- OpenOffice
- PrimoPDF
- pdftk
- Microsoft Save as PDF or XPS Add-in

Uploading PDFs
To review PDF documents, just attach 145 them to the review as you would any other file and when you open the diff viewer, the content area will display the documents for review.

Uploading Other Document Types Under Windows
Use Print To Review 282 to add Windows documents to a review.

Making Comments
Commenting on documents works exactly like commenting on images 162.

To comment on the documents, simply click on the document at a point where you would like your comment to appear and begin typing your comment. A blue pushpin ( ) will appear on the image to indicate the location of the currently selected conversation. Other pushpins indicate other conversations on the image:

Red Open Defects

© 2003-2011 SmartBear Software

Yellow	Unread Comment
Green	Fixed Defects or External Defects
Grey	No Defects

The pins may be turned off by clicking the pushpin toggle in the document review toolbar at the bottom of the content pane. This is useful when the pushpins cover important parts of the document.

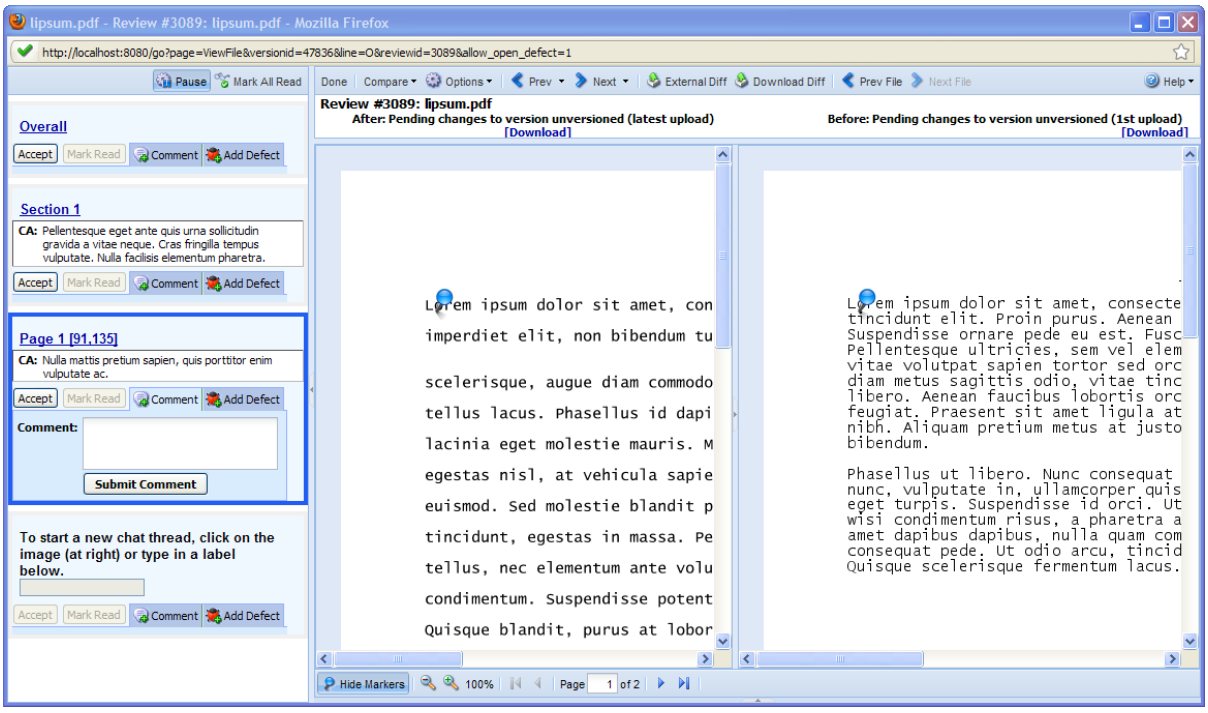
Alternatively, you may specify an arbitrary label, such as "Section 5.1", to describe the content you are referring to. Using labeled locations decreases the confusion that can occur when large changes occur in a text document resulting in the pushpins not being adjacent to the corresponding text. Label text may not be updated.

Other than the difference in location markers, review chats 167 are handled in the same manner as in reviewing text files.

Comparison

Documents can only be viewed in a **single** or **side-by-side** layout. There is no over-under view available currently for documents.

Note: PeerReview Complete does not compare document file content. Differences in text will not be highlighted, and pushpins on a previous version will remain in the same location coordinates on newer versions.

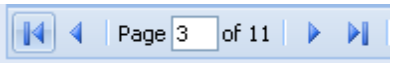

lipsum.pdf - Review #3089: lipsum.pdf - Mozilla Firefox
http://localhost:8080/go?page=ViewFile&versionid=47836&line=O&reviewid=3089&allow_open_defect=1
Pause
Mark All Read
Done
Compare
Options
Prev
Next
External Diff
Download Diff
Prev File
Next File
Help
Review #3089: lipsum.pdf
After: Pending changes to version unversioned (latest upload)
[Download]
Before: Pending changes to version unversioned (1st upload)
[Download]
Overall
Accept
Mark Read
Comment
Add Defect
Section 1
CA: Pellentesque eget ante quis urna sollicitudin gravida a vitae neque. Cras fringilla tempus vulputate. Nulla facilisis elementum pharetra.
Page 1 [91,135]
CA: Nulla mattis pretium sapien, quis porttitor enim vulputate ac.
Comment:
Submit Comment
To start a new chat thread, click on the image (at right) or type in a label below.
Lorem ipsum dolor sit amet, con
imperdiet elit, non bibendum tu
scelerisque, augue diam commodo
tellus lacus. Phasellus id dapi
lacinia eget molestie mauris. M
egestas nisl, at vehicula sapie
euismod. Sed molestie blandit p
tincidunt, egestas in massa. Pe
tellus, nec elementum ante volu
condimentum. Suspendisse potent
Quisque blandit, purus at lobor
Lorem ipsum dolor sit amet, consecte
tincidunt elit. Proin purus. Aenean
Suspendisse ornare pede eu est. Fusc
Pellentesque ultricies, sem vel elem
vitae volutpat sapien tortor sed orc
diam metus sagittis odio, vitae tinc
libero. Aenean faucibus lobortis orc
feugiat. Praesent sit amet ligula at
nibh. Aliquam pretium metus at justo
bibendum.
Phasellus ut libero. Nunc consequat
nunc, vulputate in, ullamcorper quis
eget turpis. Suspendisse id orci. Ut
wisi condimentum risus, a pharetra a
amet dapibus dapibus, nulla quam com
consequat pede. Ut odio arcu, tincid
Quisque scelerisque fermentum lacus.
Hide Markers
100%
Page 1 of 2

Side-by-Side PDF Review

© 2003-2011 SmartBear Software

Scrollbars are available for each version of the document. You can also hide the version shown on the right by clicking on the arrow in the**separator** between the two versions. You are also able to move the separator to show more or less of either version. To do this, hover the mouse over the separator until the arrows appear, and click and drag until it reaches the desired location.

Changing Pages
To change pages, use the arrows at the bottom left of the comparison screen.



The arrows will allow you to go to the **First Page of the Document**, the **Previous Page**, the **Next Page**, or the **Last Page of the document**. You can also manually enter a**page number** and hit enter to jump to any arbitrary page. The page numbers will refer to the latest upload of document shown.

Zoom
Both the side-by-side and single view support zooming the images in the display. The document review toolbar includes two magnifying glass buttons for zoom and a label displaying the current zoom scale. Zoom out is useful if the page is too large to display in the browser window without scrolling. Zoom is also useful for more precise placement of the comment markers. However, whether zoomed in or out, the comment markers still refer to pixel locations in the image at 100% zoom, so subpixel comments are not supported. If a zoom operation would cause the currently selected location to scroll offscreen, the image will be scrolled to recenter the selected location.

Rescaling of documents is performed by rerendering the document on the PeerReview Complete server and could be slow depending on the size of the document and the selected image scale.

3.2.5.4 Reviewing URLs

URLs can also be reviewed in PeerReview Complete.

However, it is important to note certain limitations. Currently, URLs are rendered in the browser as "live." This implies that clicking on a link on the URL page will direct you to a new page. This also means that you will not be able to anchor comments in the same manner as in image and document reviews.

With our current URL review functionality, newer versions of the URL will not be displayed with older versions, meaning you cannot display multiple versions of the same web page. If you would like to compare multiple URL versions, we recommend that you take static snap shots of the web page and use the image review functionality 162 of PeerReview Complete.

Uploading URLs
To upload a URL, use the Attach URL 145 feature on the review creation page.

Making Comments
As mentioned above, you will not be able to anchor comments. Instead, can specify an arbitrary label describing a part of the page or otherwise indicating to other participants what you are commenting on. In older versions of PeerReview Complete, comments in URLs were anchored to a fake "line". Those comments will continue to be displayed, but will be sorted after the new labeled locations.

© 2003-2011 SmartBear Software

smartbear.com - Review #3094: http://smartbear.com - Mozilla Firefox
Review #3094: http://smartbear.com
Live remote URL
Overall
Header
Comment:
Submit Comment
To start a new chat thread, enter a label below.
SmartBear
Code Collaborator
Peer Code Review Tool
Code Collaborator v5.0 is here!
Free Book
Learn about Code Collaborator in a FREE Webinar
Customer Quotes
NEED ASSISTANCE?
WE CAN CALL YOU IMMEDIATELY!
Request A Call
Do you need to...?
Review code without meetings, mounds of paper, or lengthy email threads?
Share code with team members in other locations?
Capture review metrics for process improvement or to meet regulatory requirements?
Improve overall code quality in the most efficient and cost-effective way possible?
Over 35,000 developers worldwide use Code Collaborator to review code with team members locally and remotely without meetings, conference calls or print-outs.
Check out Code Collaborator today and see why our customers say they can't imagine working without it.
FEATURES & SCREENSHOTS
FEATURED WHITE PAPER
NEWS & EVENTS
CodeCollaborator is full-featured but not cluttered.
View Major Features
11 Best Practices of Peer Code Review
Download White Paper
Read More News

URL Review

3.2.6 Review Chats

Chat Controls

A few controls at the top of the chat session in the diff viewer allow you to do a few special chat-related things:

When other users chat, their messages appear immediately and automatically in your chat view -- even without having to refresh the screen. You can**Pause** and **Resume** this auto-refresh mode using the toolbar button as shown above.

Usually, you have to manually mark each chat "read" or else respond to the chat to clear the "unread chat" flag. Alternatively, you can **Mark All Read** using the toolbar button as shown above. Your system administrator is able to disable this feature, so you might not actually see this on your diff viewer.

You can also hide the chat session if you do not wish to view the chat windows in the diff viewer by clicking the double arrows.

© 2003-2011 SmartBear Software

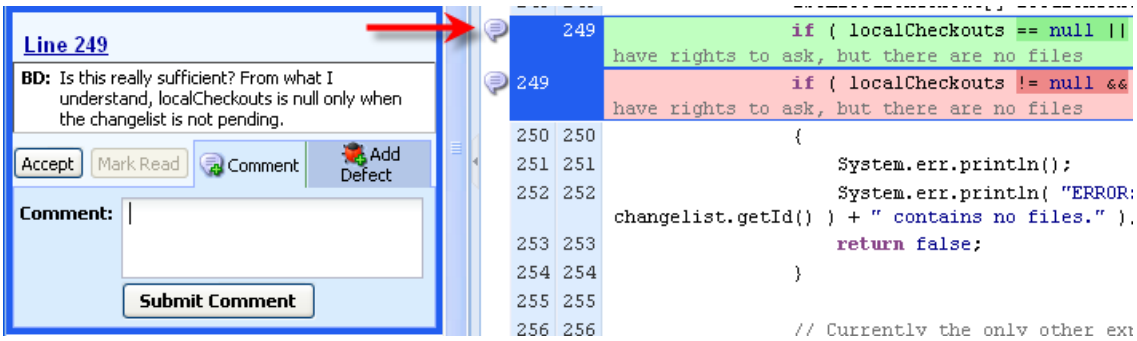
Making Conversation -- Chatting on individual lines

Viewing differences is just the start -- you need to be able to chat about specific lines and parts of a file.

To start a new chat session, just click on a line and start typing. The interface will open up automatically and accept your chat message:

To start a new chat thread, click on a line of code (at right).
Line 249
Accept
Mark Read
Comment
Add Defect
Comment:
Is this really sufficient? From what I understand, local Checkouts is null only when the changelist is not pending.
Submit Comment

When you hit `Submit Comment`, your message is transmitted to all other users and you've started a *threaded* conversation for that particular line. In this case, it would look something like this:

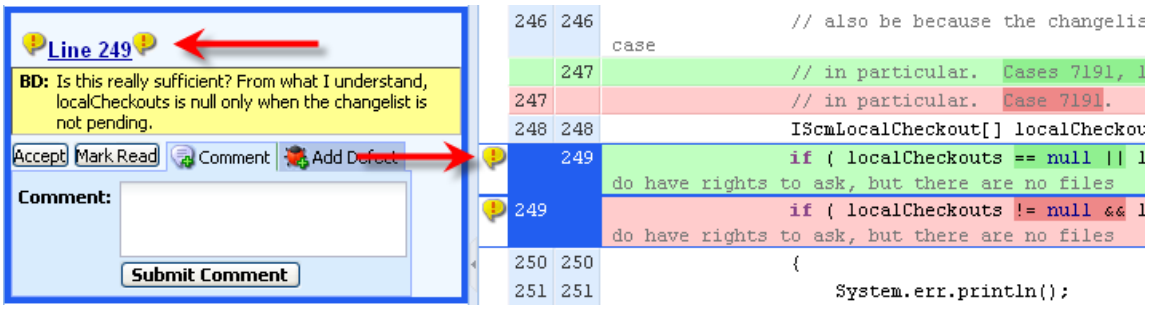


Notice how the conversation is now threaded by line (6054) and how a little icon appears next to the source where the conversation was made.

When you type the name of a file in the same review, it will turn into a link to that opens that file. You can also link to a different line number in the current file by typing "line 123".

© 2003-2011 SmartBear Software

When another user chats on that same line, that message appears in yellow and yellow bubbles help to make the conversation stand out as needing to be read. The screen updates immediately without the need for a browser "refresh":

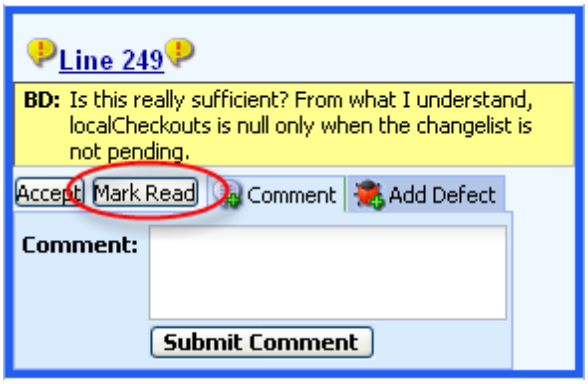

Line 249
BD: Is this really sufficient? From what I understand, localCheckouts is null only when the changelist is not pending.
Accept
Mark Read
Comment
Add Defect
Comment:
Submit Comment
246 246 // also be because the changelis
case
247 // in particular. Cases 7191, 1
247 // in particular. Case 7191.
248 248 IScmLocalCheckout[] localCheckou
249 if (localCheckouts == null || 1
do have rights to ask, but there are no files
249 if (localCheckouts != null && 1
do have rights to ask, but there are no files
250 250 {
251 251 System.err.println();

To clear this "unread chat" state, send a chat message back to the other user. You can also click the `Accept` button.

`Accept` means whatever you want it to mean. If you want, you can never use the Accept button at all. The actual effect of clicking "Accept" on a conversation is to mark that conversation read, and to put a green check mark next to that conversation in the Inspection page (main review page with list of files). Those check marks are visible to all users.

Many people use "Accept" as sort of a bookmark within the review to keep track of what files they've already looked at. Some people use it to mean explicit agreement with the conversation (as opposed to simply not replying, which is implicit agreement).

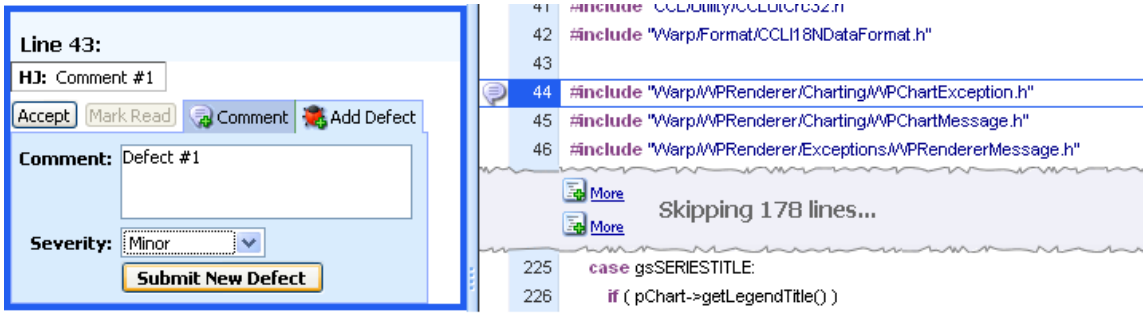
If you want to clear the "unread chat" state without saying something, click the `Mark Read` button:



© 2003-2011 SmartBear Software

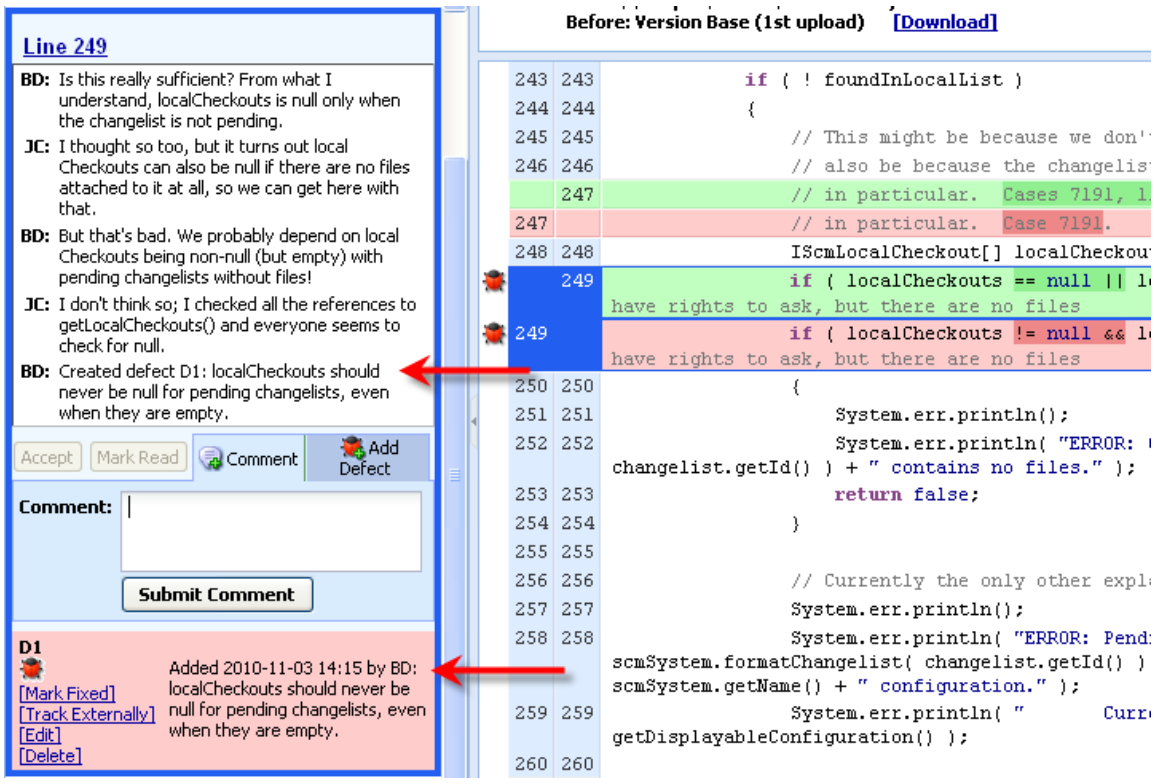
Creating Defects -- indicating that something is wrong

When you want to indicate that there's a problem that needs to be fixed -- not just more chat -- you open a defect. Do this using the `Add Defect` tab instead of the `Comment` tab:



Besides the comment text, defects can have any number of additional fields. These are allompletely configurable 104 by the system administrator and can also vary depending on thereview workflow 141 that was originally selected for the review. Because this is completely configurable, this manual cannot say exactly what the fields will be or what they mean; ask your administrator for details. It is common to see fields like severity, type, checklist item, and phase-injected.

Once the defect is created, it is added to the chat area and a defect log is started on that line:


Line 249
BD: Is this really sufficient? From what I understand, localCheckouts is null only when the changelist is not pending.
JC: I thought so too, but it turns out local Checkouts can also be null if there are no files attached to it at all, so we can get here with that.
BD: But that's bad. We probably depend on local Checkouts being non-null (but empty) with pending changelists without files!
JC: I don't think so; I checked all the references to getLocalCheckouts() and everyone seems to check for null.
BD: Created defect D1: localCheckouts should never be null for pending changelists, even when they are empty.
Accept
Mark Read
Comment
Add Defect
Comment:
Submit Comment
D1
[Mark Fixed]
[Track Externally]
[Edit]
[Delete]
Added 2010-11-03 14:15 by BD: localCheckouts should never be null for pending changelists, even when they are empty.
Before: Version Base (1st upload)
[Download]

Every defect is given a unique number; in the example above the number is 1. This number can be used when referencing the defect elsewhere.

© 2003-2011 SmartBear Software

You can have any number of defects on a single line; all will be logged into the list beneath the chat area.

You can **Edit** a defect by clicking `[Edit]` in the defect log. Or **Delete** a defect by clicking `[Delete]`. All of these activities will be logged into the chat conversation area as an audit trail of what happened.

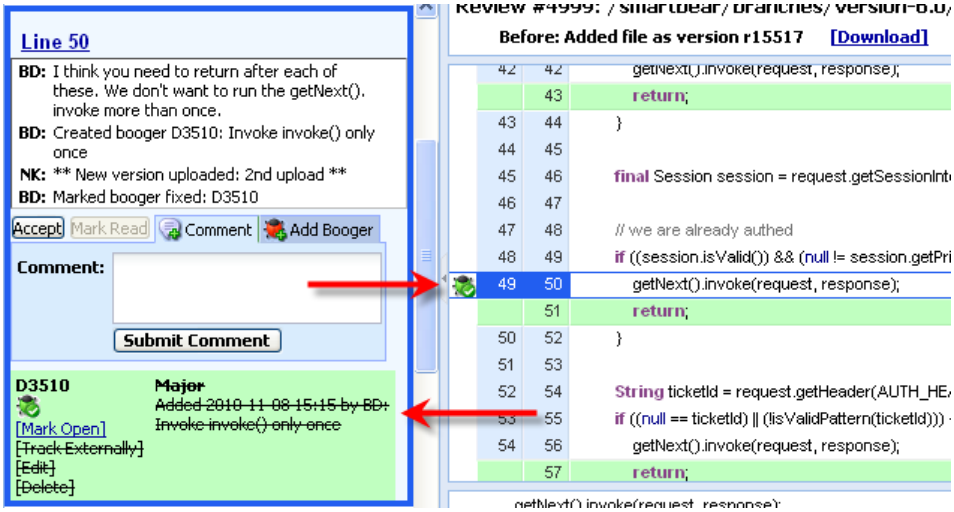
> **Warning:** It's tempting to delete a defect once the author has corrected the problem, but this is the wrong thing to do. You want to keep the defect record around, just mark it "fixed." This process is described next.
>
> Delete a defect only if it turns out that it really wasn't a defect at all.

> **Warning:** The word "defect" has many connotations that are inappropriate for peer review. This doesn't mean the problem will be mirrored in an external issue-tracking system, and it doesn't necessarily mean it was a bug! Even "bad documentation" can be a defect.
>
> A "defect" is just a way of identifying something that needs to be fixed.

Verifying that defects have been fixed

Later, after the author has attempted to fix the defects and has uploaded the new files to the review, the reviewers will verify that the fixes do fix the defect and don't open more defects in the process.

When the fix is verified, click `[Mark Fixed]` to indicate this fact. The conversation area and defect log will be updated to reflect this new state:

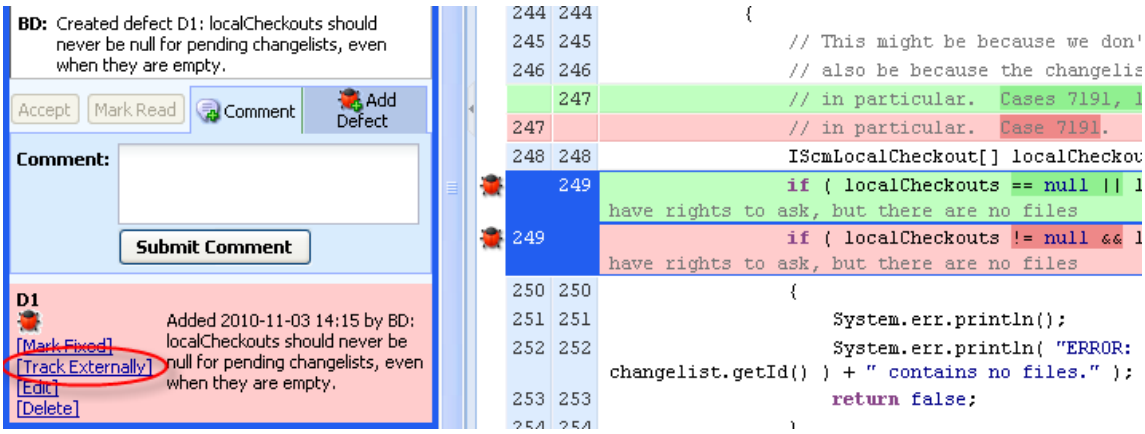


Externalizing defects

Sometimes you decide that a defect should be fixed at a later date. You don't want to mark it fixed because it isn't fixed yet. But you don't want to delete it either because it's still a defect.

© 2003-2011 SmartBear Software

The answer is to use the `[Track Externally]` link as shown below:


BD: Created defect D1: localCheckouts should never be null for pending changelists, even when they are empty.
Accept
Mark Read
Comment
Add Defect
Comment:
Submit Comment
D1
[Mark Fixed]
[Track Externally]
[Edit]
[Delete]
Added 2010-11-03 14:15 by BD: localCheckouts should never be null for pending changelists, even when they are empty.
// This might be because we don't
// also be because the changelis
// in particular. Cases 7191, 1
// in particular. Case 7191.
IScmLocalCheckout[] localCheckou
if (localCheckouts == null || l
have rights to ask, but there are no files
if (localCheckouts != null && l
have rights to ask, but there are no files
System.err.println();
System.err.println("ERROR:
changelist.getId()) + " contains no files.");
return false;

This does two things. First it prompts you to create a new defect in your external issue tracker, then allows you to type in the associated issue ID in a field. Second, it changes the state of the defect from "Open" to "Tracked Externally."

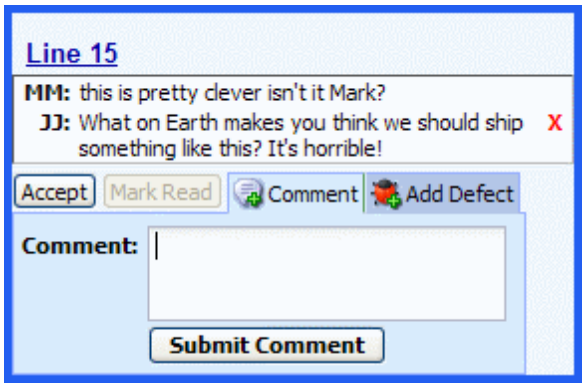
This special state tells everyone else that (a) the review can complete but (b) the problem wasn't fixed during the review and (c) how to find the external issue for further auditing. The state appears on the Review Overview 151 screen with a special icon:

You can always undo the "Tracked Externally" state by clicking on the associated link `[Not External]`. This link will be in the place where `[Track Externally]` was. This will re-open the defect, restoring it to the state it had before the externalization.

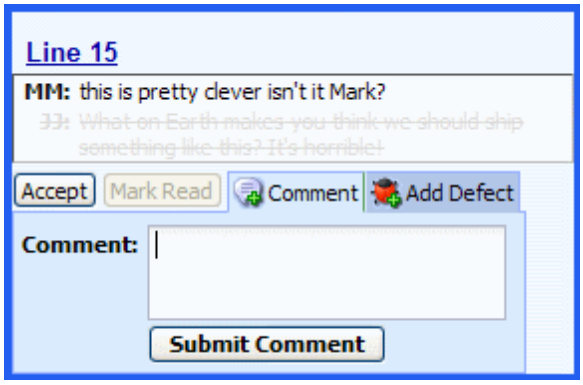
© 2003-2011 SmartBear Software

Redacting Comments

People have a knack for typing things that for various reasons they should not.



When you do this, you can redact your harsh, erroneous or otherwise inappropriate statement by clicking the **X** to the right of your comment. The comment will then be displayed in a much more difficult to read manner.



This styling is meant to deter the casual reader from actually reading the text. You cannot undo redacting a comment. If you have accidentally redacted a comment, you can copy the text from the redacted comment into a new comment. System administrators are allowed to redact any comment. Normal users are allowed to redact comments they authored.

3.2.7 FAQ's

Can I highlight a range of lines in a file?

Currently there is no way to highlight a range of lines, although this is asked for frequently.

We're going to implement this feature, but there are some subtle UI considerations that are not immediately obvious. For example, say person A highlights lines 10-20. Then B wants to highlight lines 15-25. The UI must make those ranges and how to highlight those ranges clear. Then what if C wants to comment on just line 10, the edge of A's range? The conversation threads need to make sense, the highlighting needs to represent these cases clearly, and it has to be clear how to highlight one of those or make a new comment.

© 2003-2011 SmartBear Software

This is why it's not trivial, but we are working on it.

How can I delete files in a review?
Currently, files can be deleted from a review as long as there are no comments and/or defects associated. The New Review page 143 has a link that removes uploaded changesets. One option is to cancel 148 the review and start another. Also, note that you can optionally annotate 146 uploaded files before the review is started. This step additionally allows the author a chance to verify the right files are in the review.

3.3 Searching & Reporting

Various reports 176 and a sophisticated search 174 engine help you track reviews completed, reviews in progress, and changes in version control that haven't been reviewed.

PeerReview Complete also creates metrics 456 reports for things like time spent in reviews, time between phases, kLOC/hour reviewed, defects per kLOC, time per defect, and many others.

Reports can be viewed on-line or exported in a variety of formats. Reports can also be saved and e-mailed.

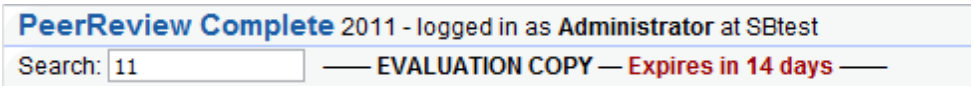
3.3.1 Searching

PeerReview Complete has a sophisticated search facility allowing you to find reviews in many ways. Uses include:

- Find reviews by review ID.
- Find reviews with substrings in title and custom fields 104.
- Find reviews with specific participants 142.
- Find reviews of particular changelists by check-in comment or changelist ID.
- Find reviews of certain files by file path.
- Find reviews having a substring in comment or defect text.

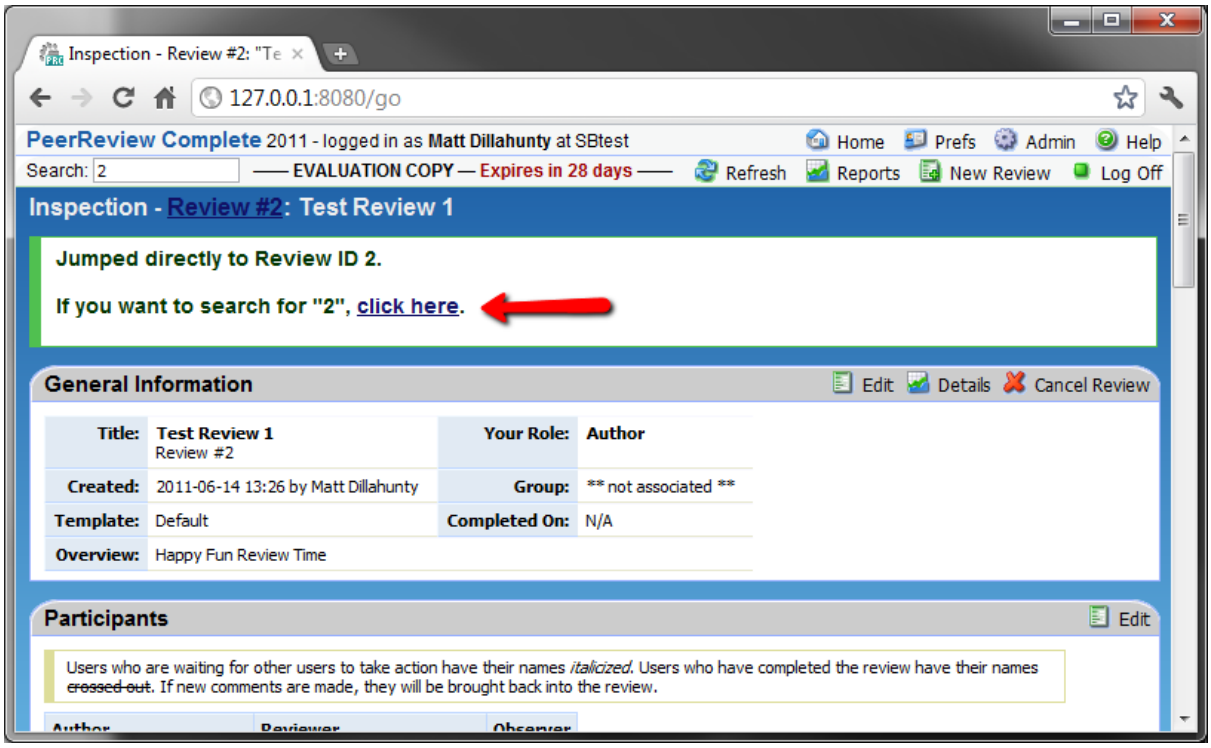
Jumping to Reviews
Our newest feature regarding the search box is the ability to jump to a review by entering the unique review ID number into the search box. If the number entered matches a review ID number, you will automatically be taken to review. This allows you to bypass search results and quickly find the searched review.

Enter the Review ID number into the search box at the left side of the website menubar:



The browser will be automatically directed to Review # 1243:

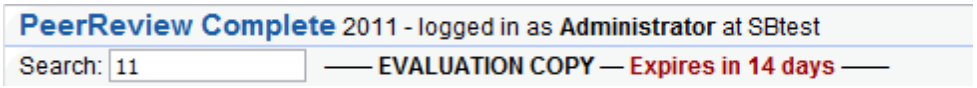
© 2003-2011 SmartBear Software


Inspection - Review #2: "Te
127.0.0.1:8080/go
PeerReview Complete 2011 - logged in as Matt Dillahunty at SBtest
Home Prefs Admin Help
Search: 2
EVALUATION COPY — Expires in 28 days
Refresh Reports New Review Log Off
Inspection - Review #2: Test Review 1
Jumped directly to Review ID 2.
If you want to search for "2", click here.
General Information
Edit Details Cancel Review
Title: Test Review 1
Review #2
Your Role: Author
Created: 2011-06-14 13:26 by Matt Dillahunty
Group: ** not associated **
Template: Default
Completed On: N/A
Overview: Happy Fun Review Time
Participants
Edit
Users who are waiting for other users to take action have their names italicized. Users who have completed the review have their names crossed out. If new comments are made, they will be brought back into the review.

There will be a box at the top which notifies you that you have jumped directly to the review with the matching ID number. If you did not intend to go to the review and wanted search results instead, click on the link indicated by the red arrow above.

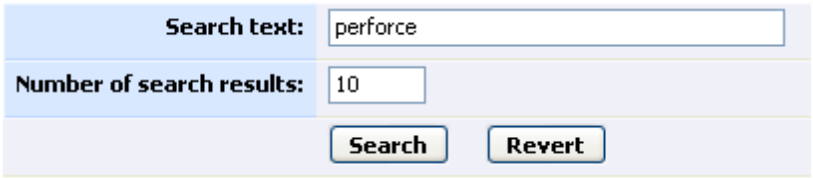
Starting a Search

To start a search, type the search term into the box at the left side of the web site menubar:



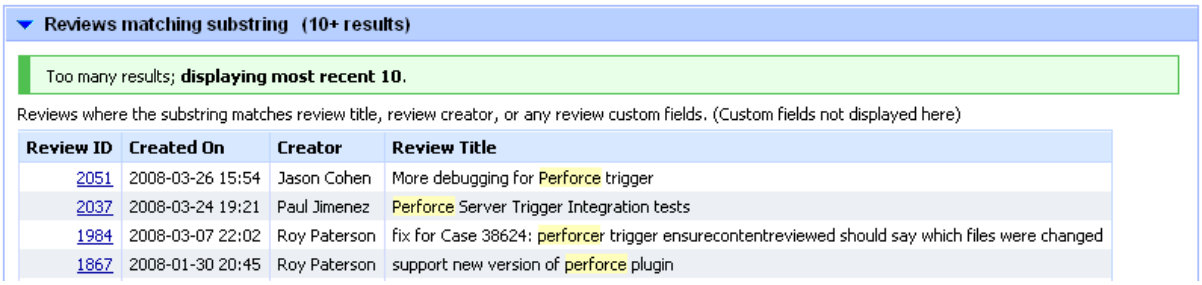
The system will automatically search in all the ways listed above.

The number of displayed search results will be limited if the search produces too many results. The default number displayed will usually be "10." This can be changed in the form that will appear at the top of the search results. To display more or less than shown, change the value in the "number of search results" field and click "search."



© 2003-2011 SmartBear Software

Because there are many types of searching, each result block is drawn separately. For example, here's the results of searching inside review title and custom field text:



Reviews matching substring (10+ results)

Too many results; **displaying most recent 10.**

Reviews where the substring matches review title, review creator, or any review custom fields. (Custom fields not displayed here)

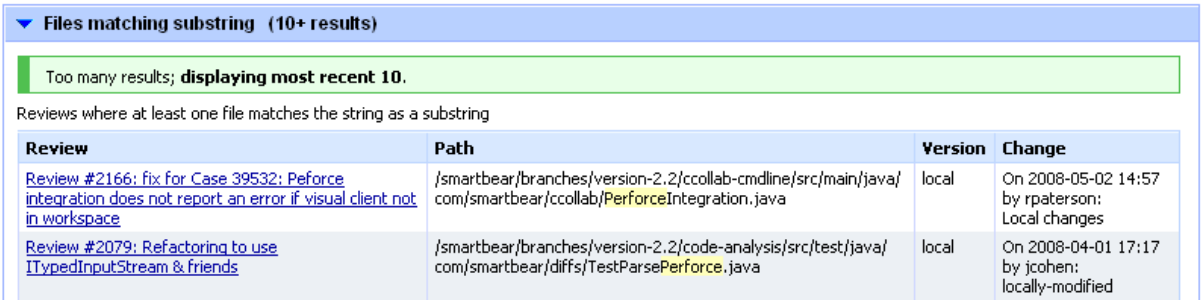
Review ID	Created On	Creator	Review Title
2051	2008-03-26 15:54	Jason Cohen	More debugging for Perforce trigger
2037	2008-03-24 19:21	Paul Jimenez	Perforce Server Trigger Integration tests
1984	2008-03-07 22:02	Roy Paterson	fix for Case 38624: perforcer trigger ensurecontentreviewed should say which files were changed
1867	2008-01-30 20:45	Roy Paterson	support new version of perforce plugin

Note several features of the search results:

- Text at top of the block explains exactly what the block is searching on.
- Reviews are displayed in creation-date order, most recent first.
- Specific search term is highlighted in yellow.
- Jump to a review by clicking the ID link.
- When there are many results, a message appears near the top in green.

You may notice that some reviews do *not* have the yellow highlight. Why did these reviews match the search? This search looks not only at the review title and participants but also in all custom fields. Those fields are not, however, displayed in this view, so nothing can be highlighted. (Custom fields are not shown because there can be a large number of them which would make the search results difficult to read.)

Here's another example where the block is searching over files present in the review:



Files matching substring (10+ results)

Too many results; **displaying most recent 10.**

Reviews where at least one file matches the string as a substring

Review	Path	Version	Change
Review #2166: fix for Case 39532: Peforce integration does not report an error if visual client not in workspace	/smartbear/branches/version-2.2/ccollab-cmdline/src/main/java/com/smartbear/ccollab/PerforceIntegration.java	local	On 2008-05-02 14:57 by rpaterson: Local changes
Review #2079: Refactoring to use ITypedInputStream & friends	/smartbear/branches/version-2.2/code-analysis/src/test/java/com/smartbear/diffs/TestParsePerforce.java	local	On 2008-04-01 17:17 by jcohen: locally-modified

3.3.2 Reporting

PeerReview Complete comes with built-in reports. The built-in reports are divided into the following groups:

- Customizable Review Reports 177
- Review Detail Reports 179

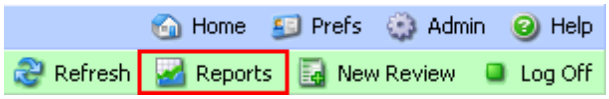
© 2003-2011 SmartBear Software

- Customizable Defect Reports 180
- Customizable User Reports 180
- List Reports 181

For more complicated reports, users have the option of writing their own custom reports 427 using an external reporting tool such as Excel, Access, Crystal Reports, or Business Objects.

Built-In Reports

To access the built-in, web-accessible reports, click the Reports menubar option:



If you are not an administrator, you may or may not have access to view reports depending on the Reports Access 76 setting.

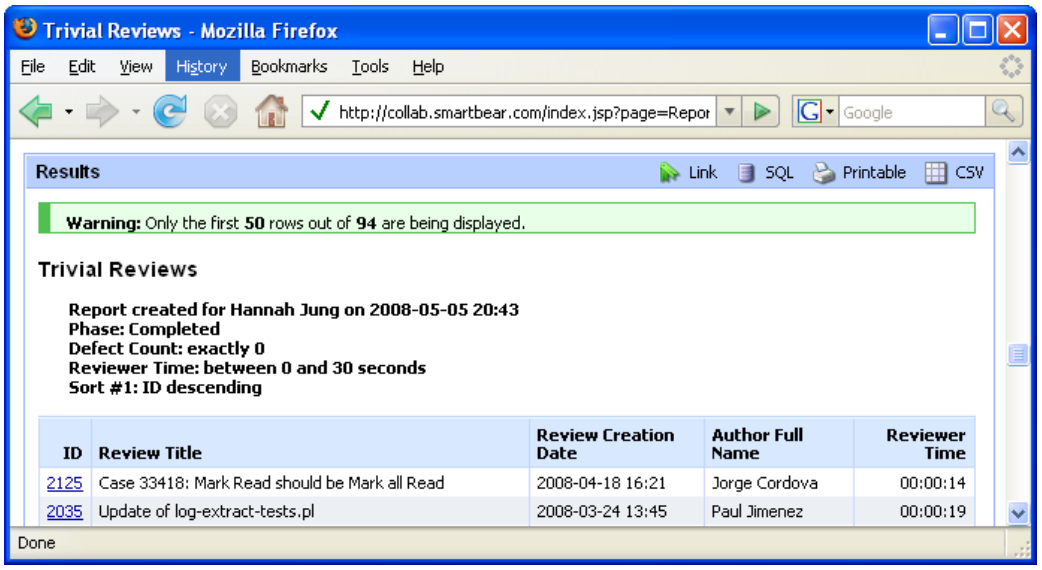
Customizable Reports

The first group of reports will be the Customizable Review Reports. Customizable Review Reports enable you to change the settings and tailor the reports to your preferences. These include the following:

Customizable Review Reports

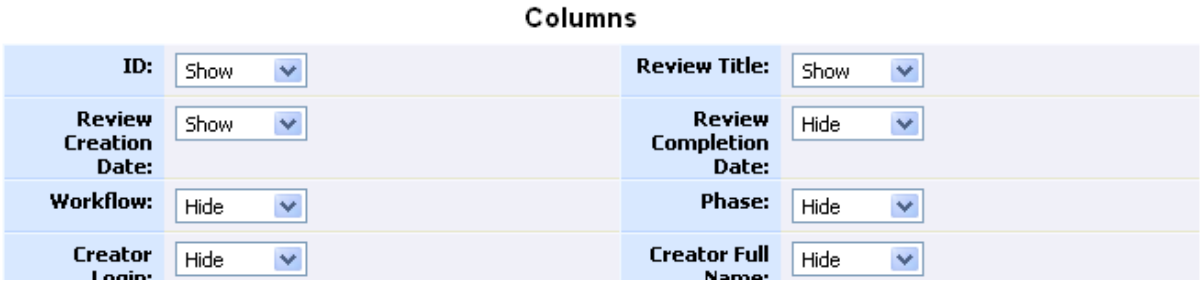
Report	
Reviews Currently In Progress	Reviews in Planning, Inspection, or Rework phases.
My Reviews, Currently In Progress	Reviews you're participating in which are currently in Planning, Inspection, or Rework phases.
Recently Completed Reviews	Reviews completed normally (e.g. not cancelled).
Stalled Reviews	Reviews where no communication has happened for a while.
Trivial Reviews	Reviews completed normally but too quickly to have been done properly.

Once you click on a report, you will be directed to another page with the chosen report on the bottom of the page. The following is the basic format of each of the customizable reports:



© 2003-2011 SmartBear Software

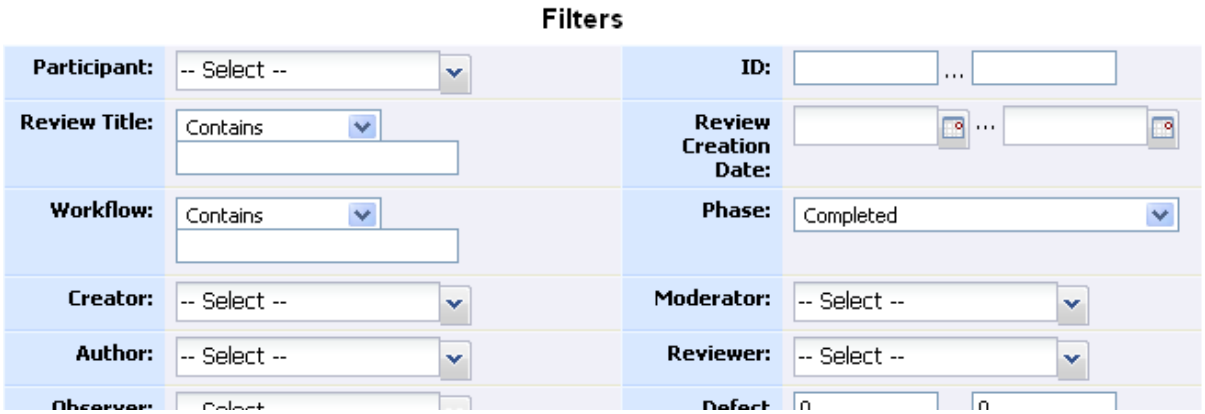
To view and edit customizable options, scroll to the top of the page.



Columns will be the first set of options given. This section gives you the option of showing or hiding certain fields, allowing you to choose how much or how little you view in the report. Note that the fields may not be identical to the example given above. The quantity and content of the fields will differ depending on how your local PeerReview Complete configuration is set up.

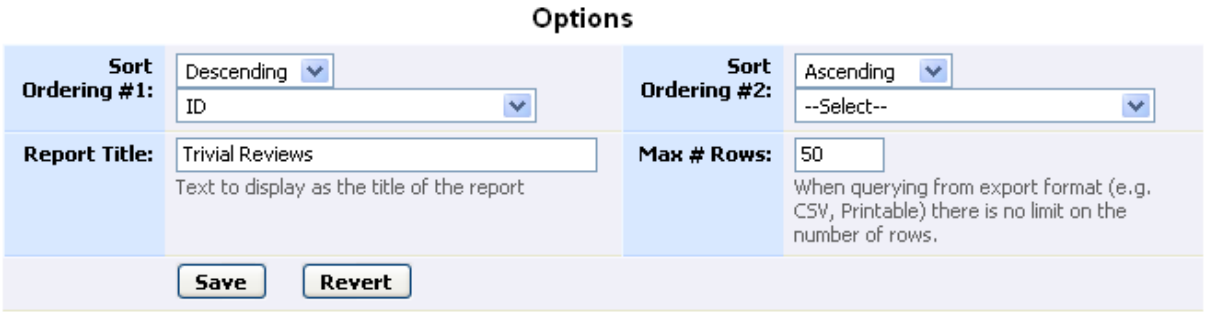
Most of the field names are self-explanatory, but here are some definitions to those that are not:

- LOC - Total number of lines of code
- LOC Added - Lines added
- LOC Removed - Lines deleted
- LOC Modified - Lines with modifications
- LOC Delta - Change in line count, i.e., current - previous
- LOC Changed - Also known as churn, this is lines added + removed + modified



Filters will be the second set of customizable fields given. Use this section to filter out your search results. Again, your fields will reflect your local PeerReview Complete configuration.

© 2003-2011 SmartBear Software



The **Options** sections allows you to edit the sorting, title, and number of rows of the review. It should be noted that the report has a limit on the maximum number of rows that can be viewed. You can change the number in the "Max # Rows" field, and all rows will be visible when you use export to SQL, CSV, or print your resulting report.

Once you've modified the report to your liking, you can use the toolbar on the top right of the Results box to export your results:

Clicking **Link** will bring up a pop-up window with a hyperlink that will take you to the appropriate link to the customized review settings. You can bookmark this link to use whenever you want to revert back to the review configuration.

The **SQL** lets you view your report in an SQL query to put into your own custom reporting tool.

The **Printable** button will prepare the report in a printable format.

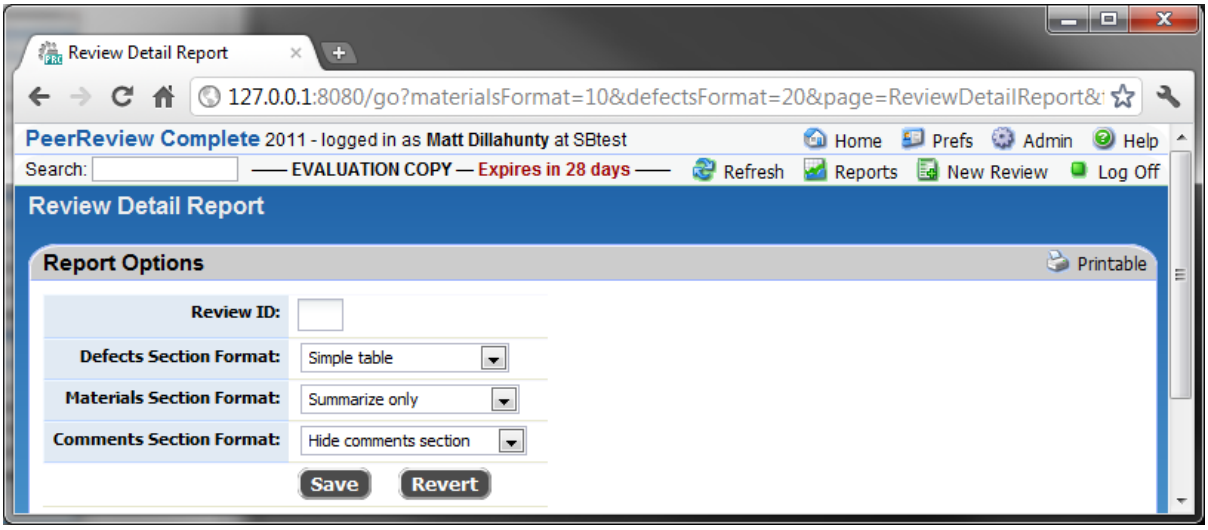
You can also download the report into **CSV**. Note the **CSV** output format does not consume a license.

Review Detail Reports

Review Detail Reports are reports on single reviews. You can choose pre-set reviews between a brief summary or a detailed report on a review.

Once you click on a report, you will be given the following form:

© 2003-2011 SmartBear Software



Enter the unique review ID number, and change the default settings if you wish. You can set the format fields to expand on or abridge the report. Clicking "Save" will generate the report below the report options. An Overview and Participants list will always be given, even when all options are set to hidden. Defects, Materials, and Comments will be shown depending how you configure the format fields.

Review Detail Reports are also accessible through the Review Summary Screen 149.

Customizable Defect Reports

Customizable Defect Reports are formatted in the same manner as Customizable Review Reports 177, but these reports are centered around defects. They include the following reports:

Customizable Defect Reports

Report	
All Defects	List of all defects from all reviews
My Recent Defects	List of defects you created recently
My Open Defects	List of defects you created and that are still open

User Reports

User Reports are formatted like Customizable Review Reports 177, but these reports are centered around Users. They include the following reports:

User Reports

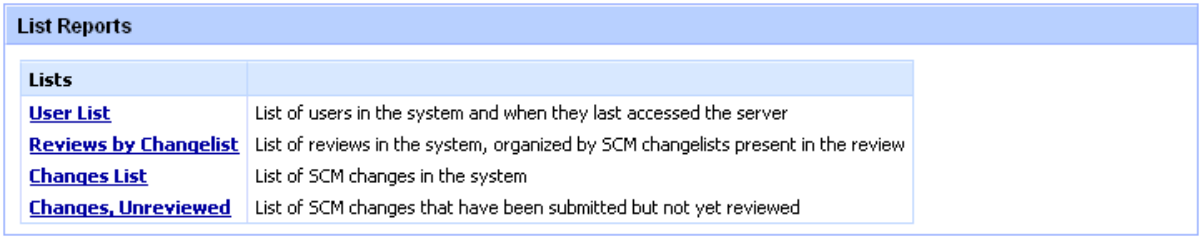
Report	
Enabled Users	List of users who are allowed to log into Code Collaborator
Disabled Users	List of users who are no longer allowed to log into Code Collaborator
Administrators	List of active administrator accounts
All Users	List of users that have been created in the system
Activity This Week	List of users who were active in reviews this week
Activity Last Week	List of users who were active in reviews last week
User Detail Report	Detailed report of user information

© 2003-2011 SmartBear Software

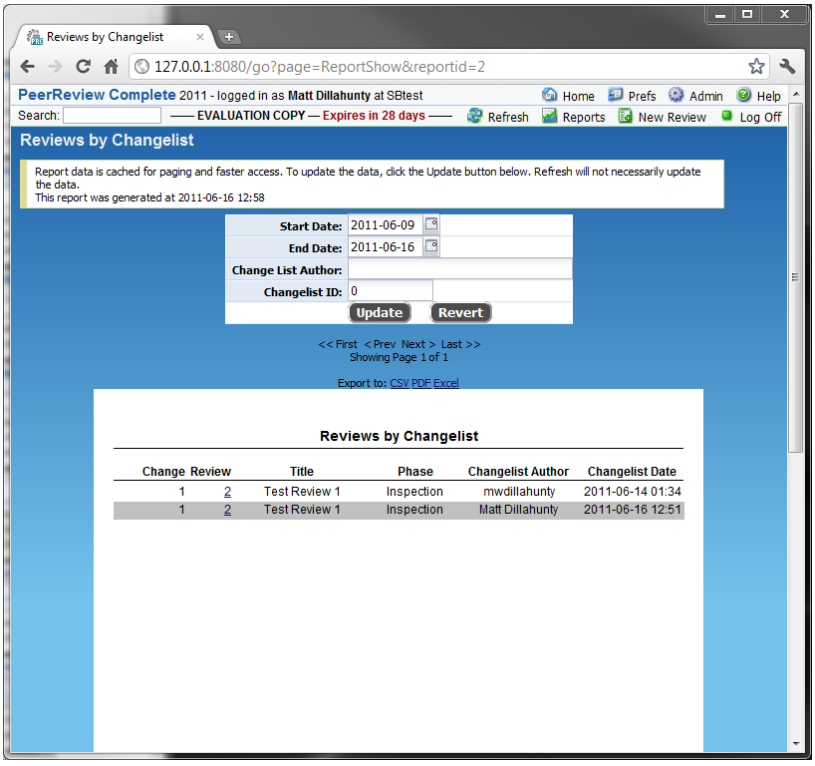
List Reports

List Reports do not give the option to be customized extensively, but these may be helpful in giving a basic coverage of reporting variables.

The first section under List Reports is Lists. These will give lists of users, reviews, defects, or SCM changes in a table.


List Reports
Lists
User List
List of users in the system and when they last accessed the server
Reviews by Changelist
List of reviews in the system, organized by SCM changelists present in the review
Changes List
List of SCM changes in the system
Changes, Unreviewed
List of SCM changes that have been submitted but not yet reviewed

Click on a list report to view it. Each list report has the same basic structure:


Reviews by Changelist
127.0.0.1:8080/go?page=ReportShow&reportid=2
PeerReview Complete 2011 - logged in as Matt Dillahunty at SBtest
Home Prefs Admin Help
Search:
EVALUATION COPY — Expires in 28 days
Refresh Reports New Review Log Off
Reviews by Changelist
Report data is cached for paging and faster access. To update the data, click the Update button below. Refresh will not necessarily update the data.
This report was generated at 2011-06-16 12:58
Start Date: 2011-06-09
End Date: 2011-06-16
Change List Author:
Changelist ID: 0
Update
Revert
<< First < Prev Next > Last >>
Showing Page 1 of 1
Export to: CSV PDF Excel
Reviews by Changelist
Change Review Title Phase Changelist Author Changelist Date
1 2 Test Review 1 Inspection mwdillahunty 2011-06-14 01:34
1 2 Test Review 1 Inspection Matt Dillahunty 2011-06-16 12:51

At the top of the page, you will have different fields depending on the report you choose. Filters may include start and end dates that you can edit to include results from a certain time period, review ID to specify a single review, and a couple other display options.

You can also export list reports into CSV, PDF, or Excel. Just click the appropriate link under the filters.

Still, most customers end up makingcustom reports 427.

© 2003-2011 SmartBear Software

Warning: Report data is cached by the server automatically. This makes it fast to page through data, but can be confusing since data isn't updated automatically when you just refresh the page.

To "really" refresh data, click the `Update` button under the list of filters.

4 Clients

PeerReview Complete has a variety of clients for creating reviews and uploading files for review. While the PeerReview Complete web server user interface provides the capability for uploading documents and, for some SCM integrations, the capability for creating reviews from committed changes, a client simplifies and expedites this process. Also, importantly, a client allows reviews to be created from local modifications to SCM files that are not accessible on the SCM server (i.e., PeerReview Complete clients allow for pre-commit reviews).

Client/Server Version Compatibility
As a general rule, older clients will work with a newer version of the server, but clients will not work with servers that are older than the client. However, it is always recommend to use the same client version as the server if at all possible.

SCM Integration
All PeerReview Complete clients integrate with SCM systems, for identifying files under source control and ease of selecting modified files to be uploaded to the PeerReview Complete server for a review. Files are uploaded as changelists. For SCM systems that don't implicitly support changelists, the clients have mechanisms for selecting a group of files to be uploaded to a review.

Creating Reviews and Uploading Files For Review
Each client also provides a means for creating a review, and for attaching a changelist to an existing review.

Other Features
Additional features are available depending on the client. The client right for your development environment may be determined by your SCM system, or just a matter of personal choice.

4.1 Installation

The Command Line Client, GUI Client, External Diff Viewer launcher, and Tray Notifier are all included in a single installer. The client installer is cross-platform and can be run in an interactive GUI mode or silently from the command-line.

Getting the Installer
Go to our website to get the latest installer for your platform:

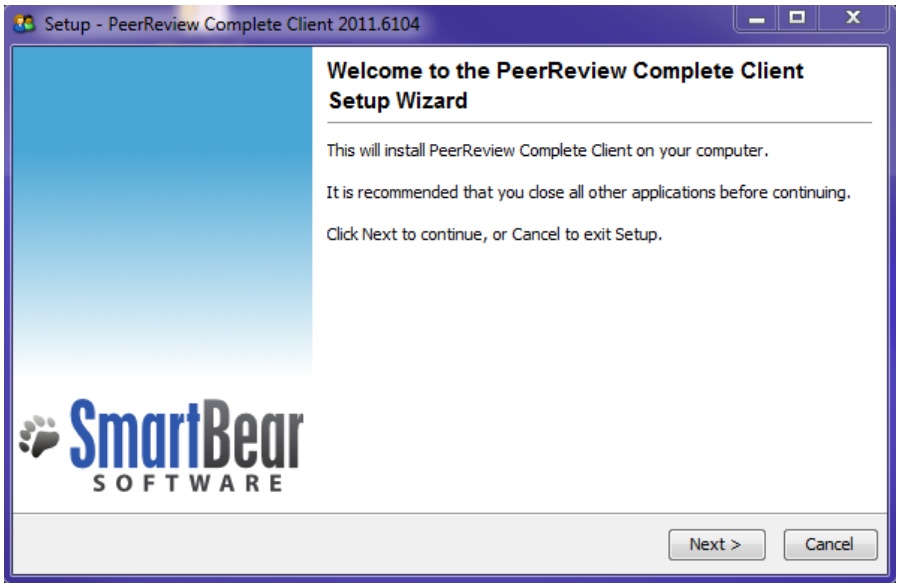
http://downloads.smartbear.com

© 2003-2011 SmartBear Software

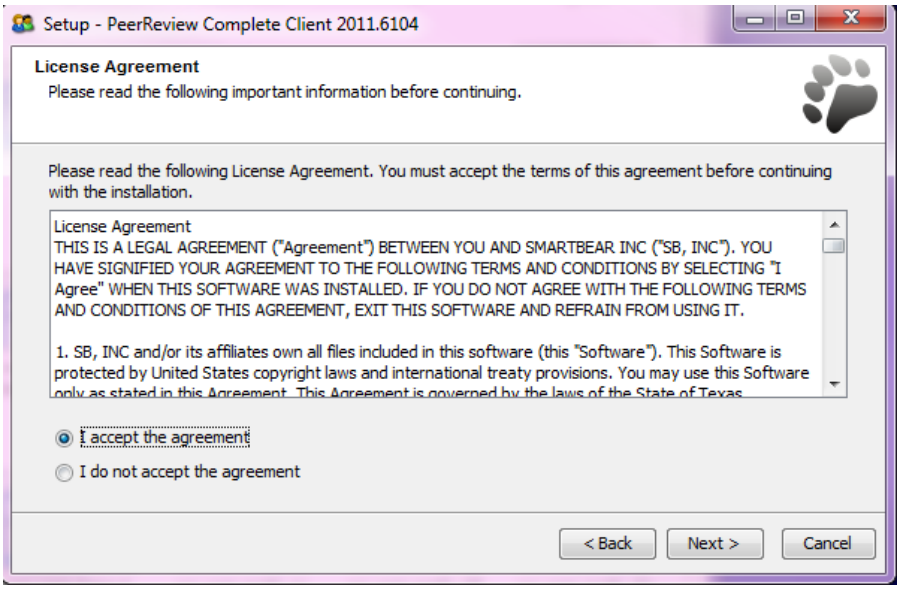
Installers are provided for Windows, Linux/Solaris, RedHat RPM, and Mac OS X. We also supply the raw files without a managing installer in the form of a compressed tar archive.

Note: If you have previously installed the client with the 32 bit installer and are now wanting to run the 64 bit installer, you must first uninstall the 32 bit client. This will not remove any configuration data you've given the client - i.e. preferences, SCM configs, and default browsers.

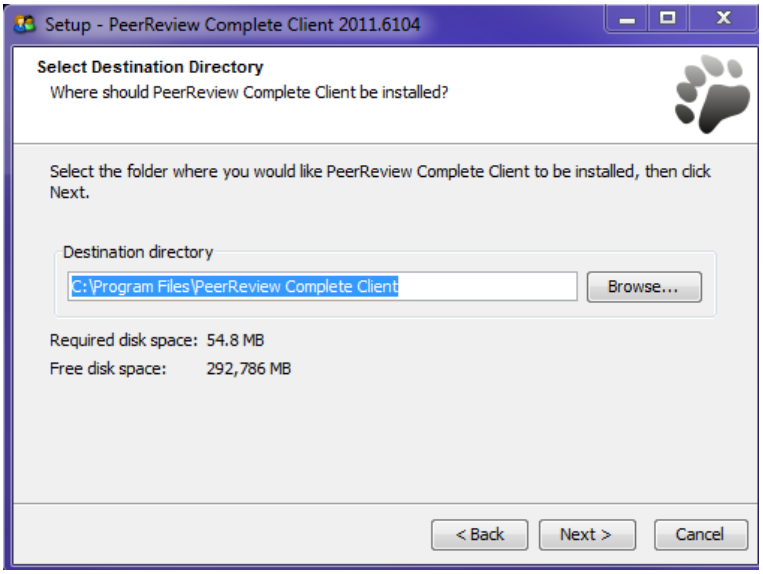
Graphical Installation
This is the recommended way to install the client software because you get a chance to see all the configuration options and allow the installer to validate your configuration.


Setup - PeerReview Complete Client 2011.6104
Welcome to the PeerReview Complete Client Setup Wizard
This will install PeerReview Complete Client on your computer.
It is recommended that you close all other applications before continuing.
Click Next to continue, or Cancel to exit Setup.
SmartBear SOFTWARE
Next >
Cancel

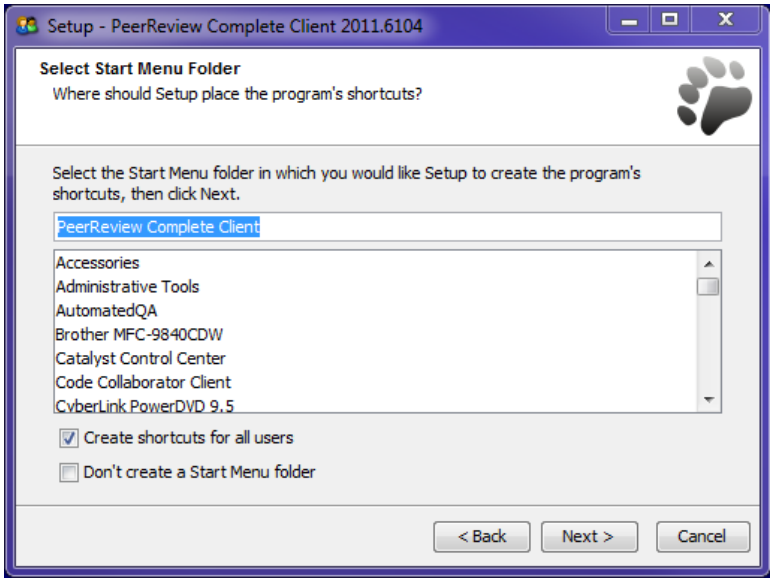
After the welcome screen, you are prompted with the EULA (End User License Agreement):


Setup - PeerReview Complete Client 2011.6104
License Agreement
Please read the following important information before continuing.
Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.
License Agreement
THIS IS A LEGAL AGREEMENT ("Agreement") BETWEEN YOU AND SMARTBEAR INC ("SB, INC"). YOU HAVE SIGNIFIED YOUR AGREEMENT TO THE FOLLOWING TERMS AND CONDITIONS BY SELECTING "I Agree" WHEN THIS SOFTWARE WAS INSTALLED. IF YOU DO NOT AGREE WITH THE FOLLOWING TERMS AND CONDITIONS OF THIS AGREEMENT, EXIT THIS SOFTWARE AND REFRAIN FROM USING IT.
1. SB, INC and/or its affiliates own all files included in this software (this "Software"). This Software is protected by United States copyright laws and international treaty provisions. You may use this Software
I accept the agreement
I do not accept the agreement
< Back
Next >
Cancel

The next screen lets you select where to lay down the files for the installation:

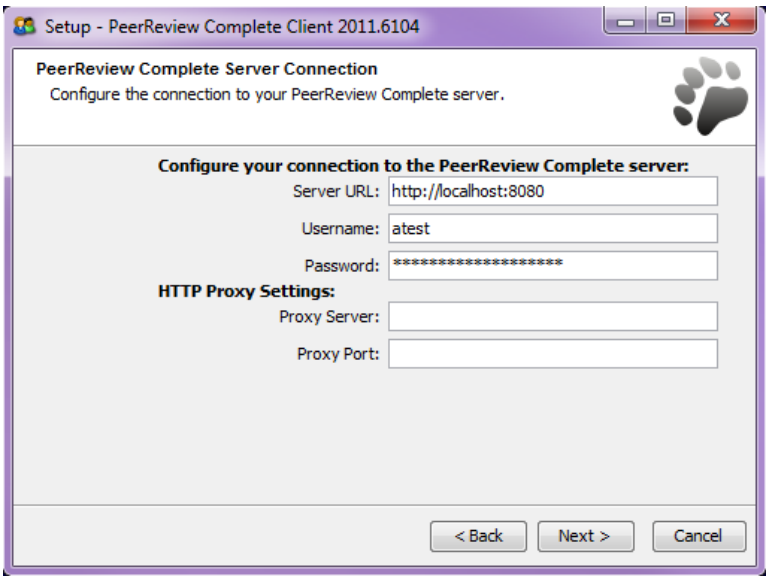

Setup - PeerReview Complete Client 2011.6104
Select Destination Directory
Where should PeerReview Complete Client be installed?
Select the folder where you would like PeerReview Complete Client to be installed, then click Next.
Destination directory
C:\Program Files\PeerReview Complete Client
Browse...
Required disk space: 54.8 MB
Free disk space: 292,786 MB
< Back
Next >
Cancel

The next screen lets you configure how and if your Start Menu (Windows only) is modified. This doesn't affect the behavior of the command-line client itself:


Setup - PeerReview Complete Client 2011.6104
Select Start Menu Folder
Where should Setup place the program's shortcuts?
Select the Start Menu folder in which you would like Setup to create the program's shortcuts, then click Next.
PeerReview Complete Client
Accessories
Administrative Tools
AutomatedQA
Brother MFC-9840CDW
Catalyst Control Center
Code Collaborator Client
CyberLink PowerDVD 9.5
Create shortcuts for all users
Don't create a Start Menu folder
< Back
Next >
Cancel

© 2003-2011 SmartBear Software

Next, you configure your connection to the PeerReview Complete server:



When you configure the URL, be sure to remember to specify the protocol (e.g.`http://` or `https://`) and port number, and path. Leaving out one of those things is the most common mistake.

Your username and password must already exist [84] on the server. Log into [125] the server first from a web browser to test your account information.

If you use a proxy, supply your proxy information as well.

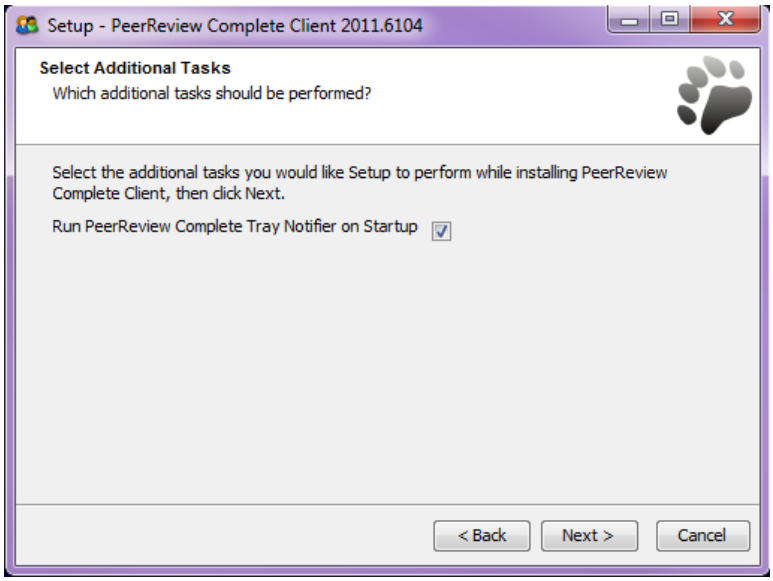
If the connection to the server cannot be established, you'll get an error dialog and an explanation of the problem:

Setup
Invalid username (cbass) or password.
Continue with current setting or retry?
If you continue you will be prompted for login configuration the first time you try to upload changes.
Note: If you are using HTTPS and the server certificate is not valid, you will need to continue and run 'ccollab login' from the command line.
Continue
Retry

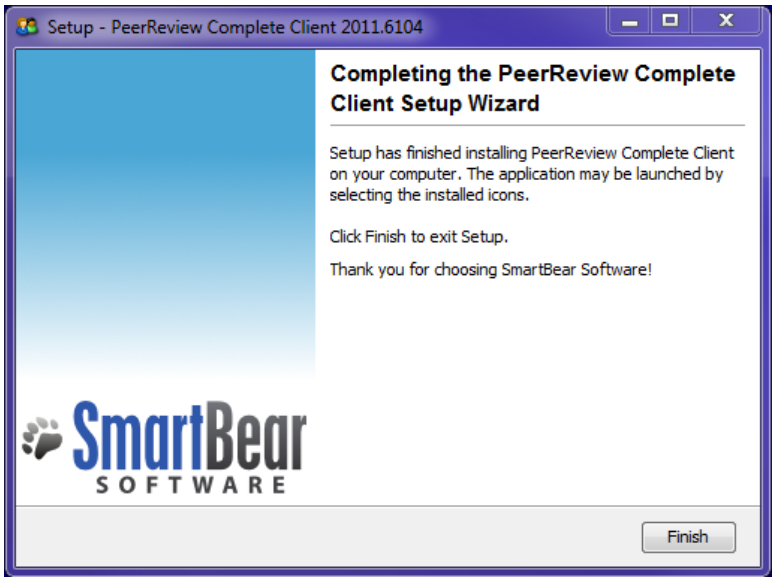
If you select `Retry`, you can supply different configuration information and try again. Otherwise, if you select `Continue`, the installation will continue but you will still need to set up your configuration [197].

After this step, you will be prompted for additional tasks. On Windows you will have the option to configure the PeerReview Complete Tray Notifier to run on Startup. On all platforms, there will be an option to configure the integrations in the Perforce Visual Tools. If this option is selected, the installer will attempt to configure the P4V and P4Win integrations [381] for the current user. Other users will still need to configure the integrations manually.

© 2003-2011 SmartBear Software


Setup - PeerReview Complete Client 2011.6104
Select Additional Tasks
Which additional tasks should be performed?
Select the additional tasks you would like Setup to perform while installing PeerReview Complete Client, then click Next.
Run PeerReview Complete Tray Notifier on Startup
< Back
Next >
Cancel

Once you click "Next," the installer will complete:


Setup - PeerReview Complete Client 2011.6104
Completing the PeerReview Complete Client Setup Wizard
Setup has finished installing PeerReview Complete Client on your computer. The application may be launched by selecting the installed icons.
Click Finish to exit Setup.
Thank you for choosing SmartBear Software!
SmartBear
SOFTWARE
Finish

Unattended Installation

To run the installer without a GUI, run it from the command-line using the `-q` switch. In this case, you can also use the `-dir [directory]` switch to specify the target install directory. The -q switch gives you a silent install and will not prompt you for any installation instructions. If you'd like to be prompted for installation instructions without using the graphical installer, you can run the installer from the console using the -c command.

This installation technique can also be used to install clients remotely on many workstations.

© 2003-2011 SmartBear Software

Warning: When run in unattended mode, the installer cannot verify the client's connection with the PeerReview Complete server. You might have to manually configure the connection with the server.

RPM Installation
The RPM can be installed using an RPM manager or from the command-line like this:

```
rpm -i ccollab_client_6_5_6500_linux.rpm
```

If you have a previous installation, you will have to remove it first using a command-line like this:

```
rpm -e ccollab_client
```

4.2 Configuration

Clients will need to be configured to both the PeerReview Complete server and the SCM server.

Your client may have been successfully configured to the PeerReview Complete server by the installer during the installation 182 process. If it hasn't been properly configured, please visit the Server Connection Configuration section 187 for detailed instructions.

Please make sure that your client is also configured to your version control or SCM. For instructions on how to do so, please visit the configuration sections under the appropriate client:

GUI Client SCM Configuration 191

Command Line Client SCM Configuration 197

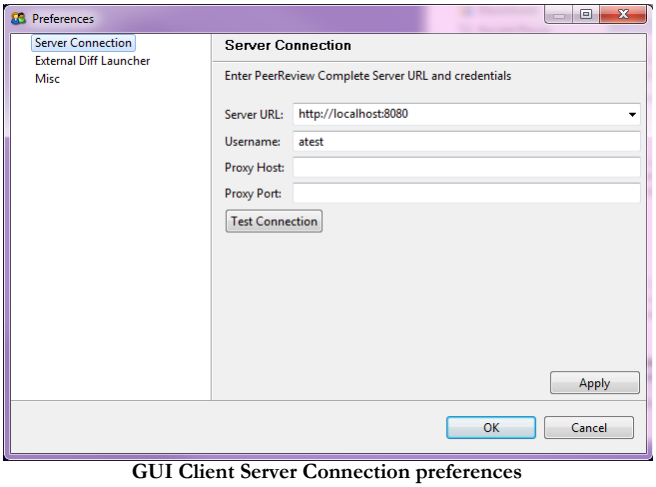
4.2.1 Server Connection Configuration

If you used the graphical installer 183, your connection to the PeerReview Complete Server should be configured already. Otherwise, you will be prompted when you try to connect.

The GUI Client, Command Line Client, Tray Notifier, and SCM Triggers share a common server connection configuration file 188. The server connection configuration can be set with a graphical interface 187 using GUI Client or the Tray Notifier, or on the command-line 188 using the Command Line Client.

Graphical Server Connection Preferences
Selecting *File -> Preferences* on the GUI Client main screen 190 or selecting *Preferences* in the Tray Notifier context menu 279 open the graphical Server Connection preferences page.

© 2003-2011 SmartBear Software



GUI Client Server Connection preferences

The *Server URL* must include the correct port number and path if applicable. The *Username* and *Password* are the same as you use when logging into the web server [125]. The proxy settings should be used if you have a client proxy between your workstation and the server.

Use the *Test Connection* button to make sure the connection is working. If it fails, the error message will be helpful.

Command-line Server Connection Global Options

The Command Line Client uses the following global options [204] to specify the connection to the PeerReview Complete Server:

- url [205] - URL to server (must be a valid, fully-specified URL)
- user [205] - Username (same as web user login [125])
- password [205] - Password (same as web user password [125])
- server-proxy-host [205] - URL of proxy server host name, if any
- server-proxy-port [205] - Port number to use on proxy server, if any

Use the `ccollab login` [214] command to connect to the PeerReview Complete Server and save your server connection options to the configuration file.

You can try testing your configuration [197] to verify the configuration is working.

Configuration Files

PeerReview Complete uses several configuration files to store configuration. When a user sets server connection configuration using the graphical interface [187] or the command-line [188], the settings are stored inside a directory called `.smartbear` inside the user's home directory. (Under Windows, the "home directory" is your "`Documents and Settings`" Profile directory.)

The `.smartbear` directory and the configuration files therein can be placed in other locations to establish default behavior. Each of the locations is loaded in a particular order of precedence, with each successive location overriding the settings (if any) in the previous locations:

© 2003-2011 SmartBear Software

1. `INSTALLDIR/.smartbear`
   Here `INSTALLDIR` refers to the PeerReview Complete installation directory. This is typically `/opt/ccollab-cmdline` under Unix or `C:\Program Files\PeerReview Complete Client` under Windows.
   This is most useful for system-wide default settings.
2. `PROFILEDIR/.smartbear`
   Here `PROFILEDIR` refers to a directory specified using the Java property `smartbear.profile` on one of the PeerReview Complete Client executables. With this property defined, the client will look in this directory for more configuration.
   This is most useful for executable-specific default settings, such as settings just for SCM server-side triggers.
3. `USERDIR/.smartbear`
   Here `USERDIR` refers to the user's home directory (under Windows, your "`Documents and Settings`" Profile directory).
   This is the default location for configuration settings to be stored, and is useful for user-specific settings.

4.3 GUI Client

The GUI Client is a cross-platform client interface to the PeerReview Complete server. It is used for uploading files, either before or after check-in

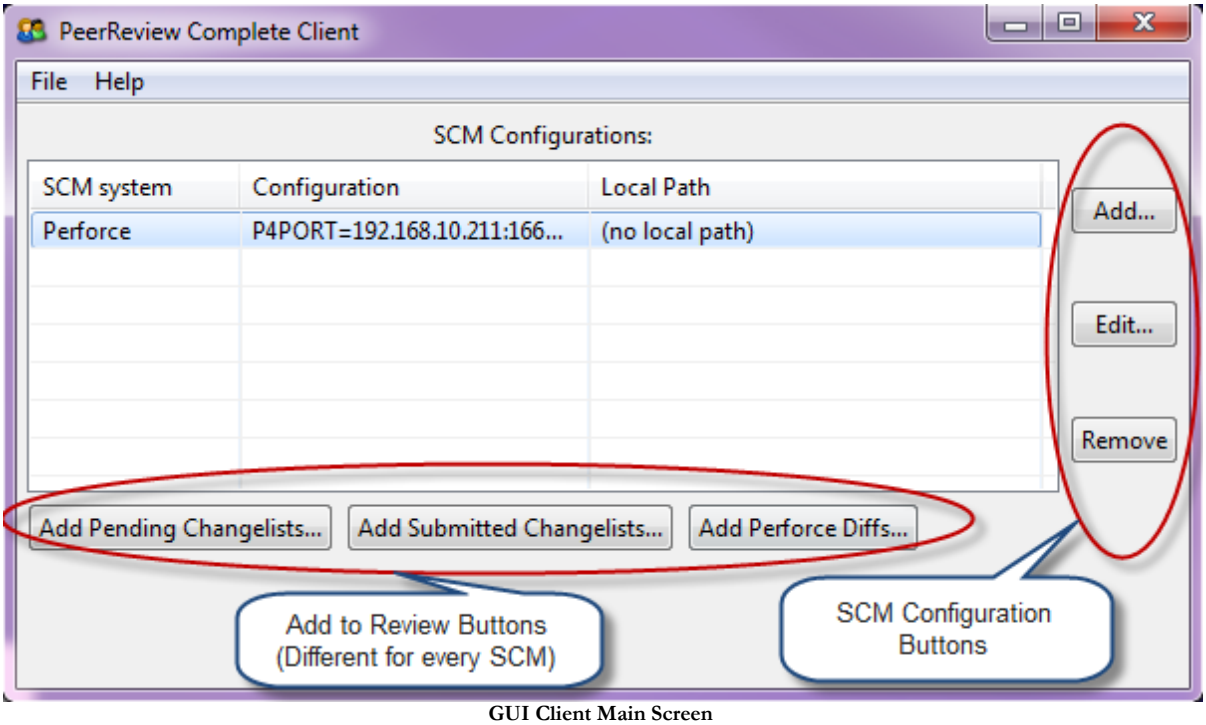
This chapter includes:

- Main Screen 190
- SCM Configuration 191
- Troubleshooting 196

© 2003-2011 SmartBear Software

4.3.1 Main Screen

Main Screen
The GUI Client's main screen contains a list of SCM configurations. Each SCM Configuration entry shows it's *SCM system*, *Configuration*, and *Local Path* (if any). Use the SCM Configuration buttons 190 on the right side of the screen to add or modify an SCM Configuration. Use the Add to Review buttons 190 on the bottom of the screen to upload review materials to the PeerReview Complete Server.



GUI Client Main Screen

Server Connection configuration
If you used the graphical installer 183, your connection to the PeerReview Complete Server should be configured already. Otherwise, you will be prompted when you try to connect. You can also select *File -> Preferences* in the system menu to open the Server Connection preferences 187 page.

SCM Configuration buttons
The buttons on the right side of the main screen are used to add or modify an SCM Configuration in the SCM Configurations list.

- *Add...* adds a new SCM Configuration
- *Edit...* edits the selected SCM Configuration
- *Remove* removes the selected SCM Configuration

Add to Review buttons
The buttons on the bottom of the main screen are used to upload materials to a new or existing Review on the PeerReview Complete Server. The buttons available will depend upon the selected SCM Configuration. Different buttons are available for different SCM systems:

© 2003-2011 SmartBear Software

- AccuRev 289
- ClearCase 324
- CVS 303
- Perforce 370
- Subversion 400
- Star Team 297
- Team Foundation 350
- Vault 393

Keyboard Shortcuts
The GUI Client also offers keyboard shortcuts as follows:

File Menu
Alt + F = File menu

Alt + P = Preferences

Alt + X = Exit

Help Menu
Alt + H = Help menu

Alt + L = Capture Debugging Log

Alt + E = Email Support...

Alt + A = About PeerReview Complete

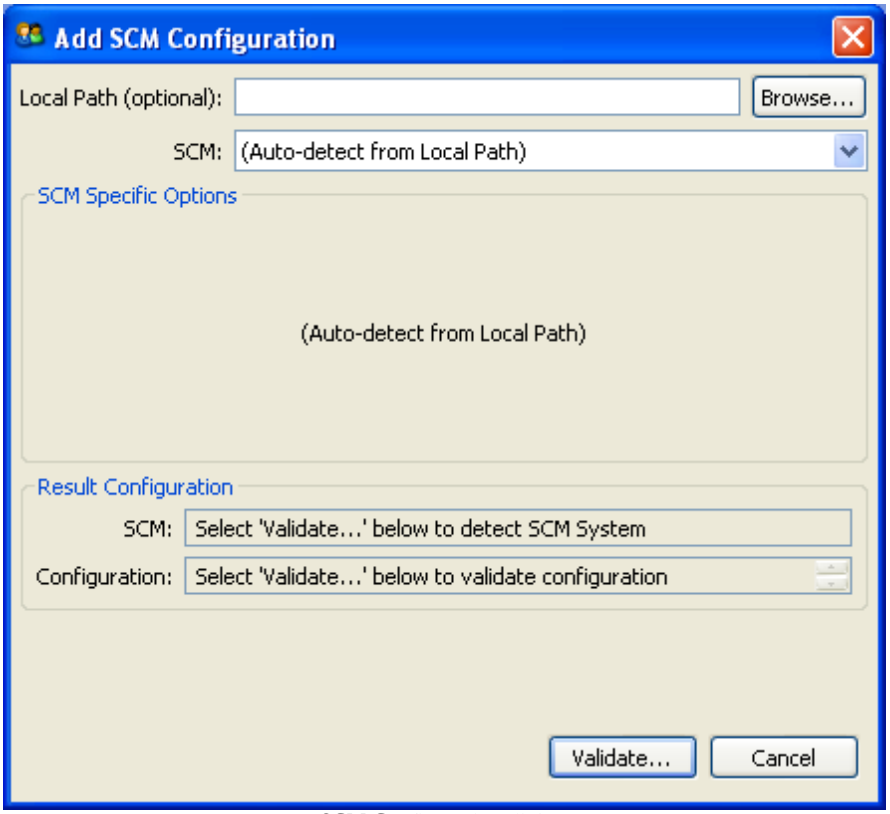
* For Mac users, 'Alt' should be replaced with 'Cmd.'

4.3.2 SCM Configuration

This section will describe how to configure the GUI Client to your SCM or version control. You will also have to configure 187 the client to the PeerReview Complete server.

Configuring Version Control
Pressing the *Add...* or *Edit...* SCM Configuration buttons 190 on the right side of the main screen 190 opens the SCM Configuration dialog.

© 2003-2011 SmartBear Software



SCM Configuration dialog

Local Path

If you have a copy of your source on your local machine, enter its location into the optional *Local Path* field using the *Browse* button. Press the *Validate...* button to detect your SCM and SCM Configuration.

Scm Specific Options

If there are additional settings necessary to connect to your SCM system, select your SCM system using the *SCM* drop-down to show the available options. There are different options available for each SCM system:

- AccuRev 289
- ClearCase 324
- CVS 303
- Perforce 369
- Subversion 399
- Vault 392

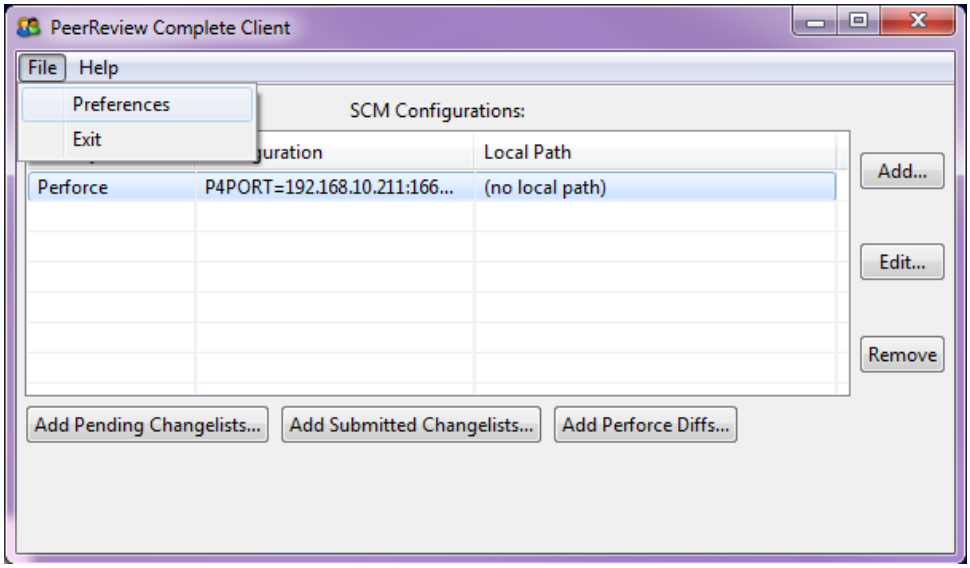
© 2003-2011 SmartBear Software

4.3.3 Preferences

The Preferences menu will allow you to change the settings for:

- Server Connection
- External Diff Viewer launcher
- Miscellaneous

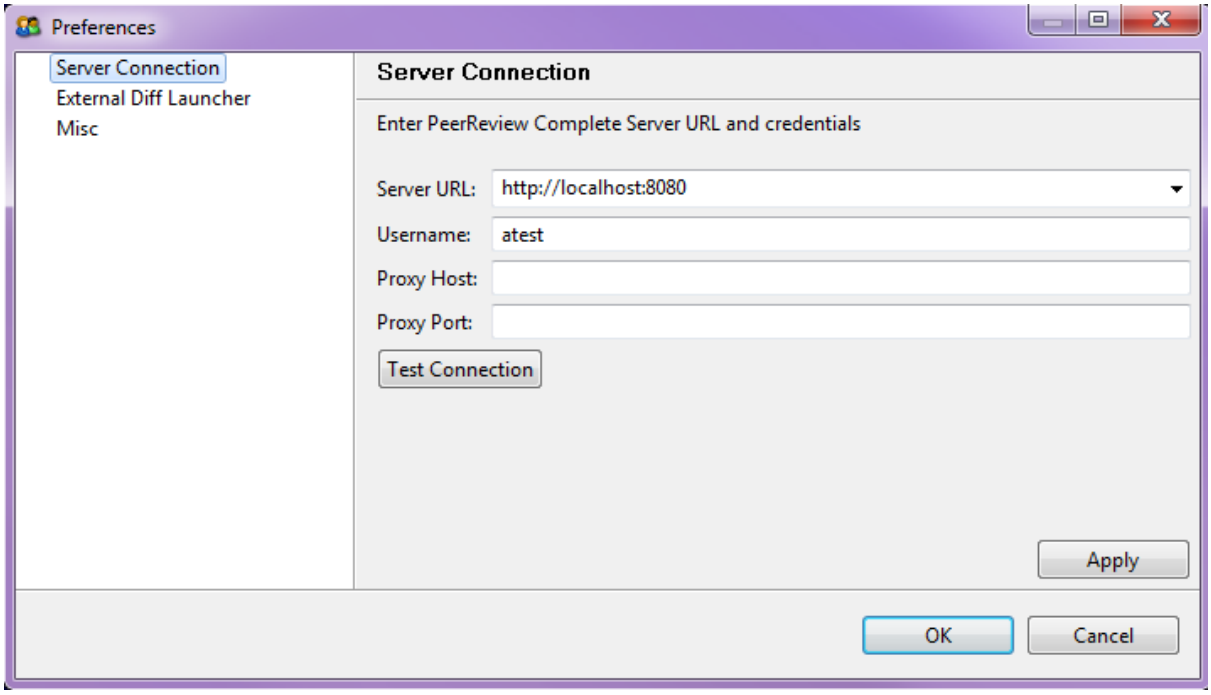
To view preferences, click on *File -> Preferences* in the main screen 278 of the GUI Client:



You can also view the Preferences menu via theContext Menu 279 of the Tray Notifier.

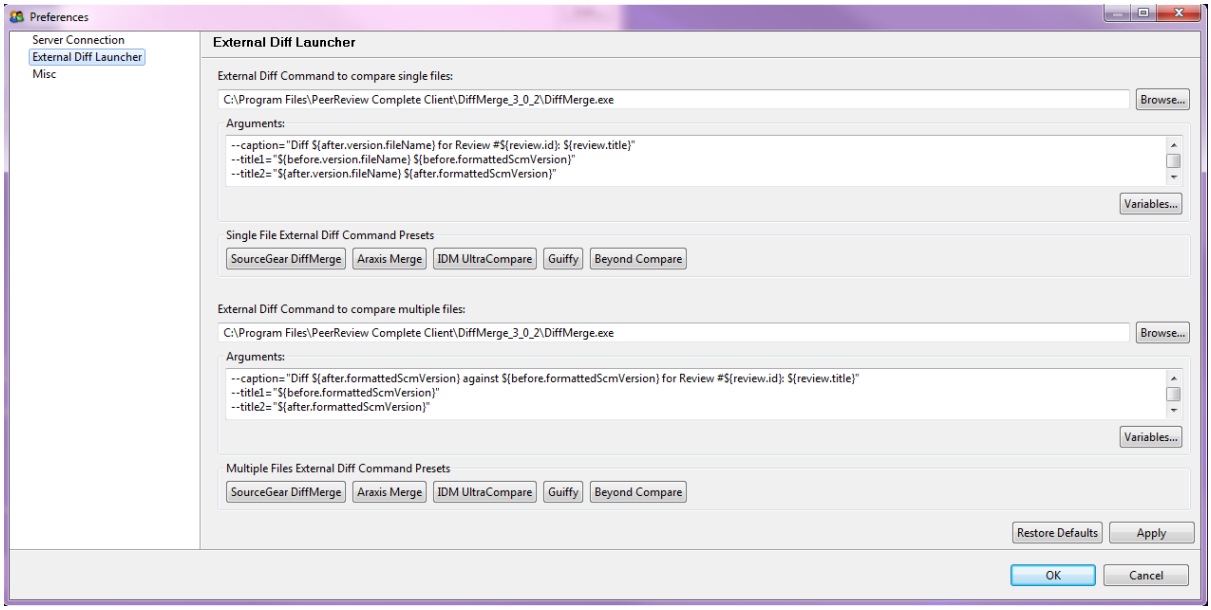
© 2003-2011 SmartBear Software

Server Connection



Here you can configure the client connection to the server.

External Diff Viewer launcher



Here you can configure settings for the External Diff Viewer launcher.

There are two separate **External Diff Command** settings, one for comparing two versions of a **single file**, and one for comparing **multiple files** (two directory trees).

© 2003-2011 SmartBear Software

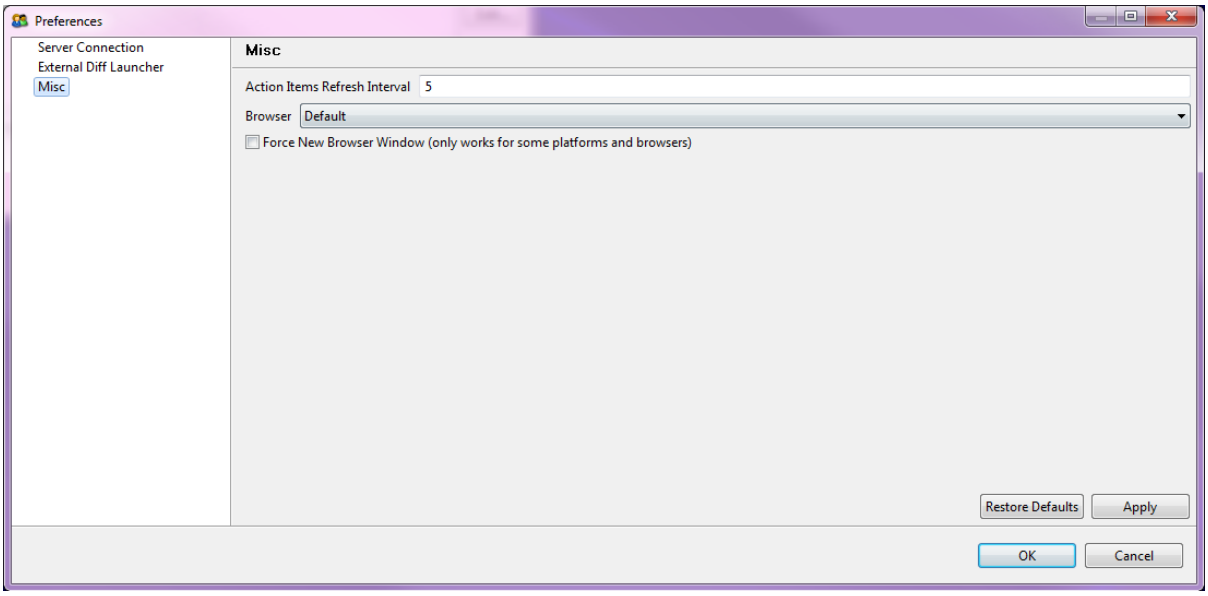
The **External Diff Command** field contains the absolute path to the executable the External Diff Viewer launcher will launch.

The **Arguments** field contains arguments passed to the executable. Certain variables specified in the **Argument** field are substituted when the external diff command is invoked. Click the `Variables` button to view the list of available variables and their descriptions.

As mentioned in the External Diff Viewer launcher section 280, we've included preset arguments to use with the following diff viewers: SourceGear DiffMerge, Araxis Merge, IDM UltraCompare, Guiffy, and Beyond Compare.

To use one of the presets, just click the appropriate button (e.g `Guiffy`) under **External Diff Command Presets**.

Miscellaneous

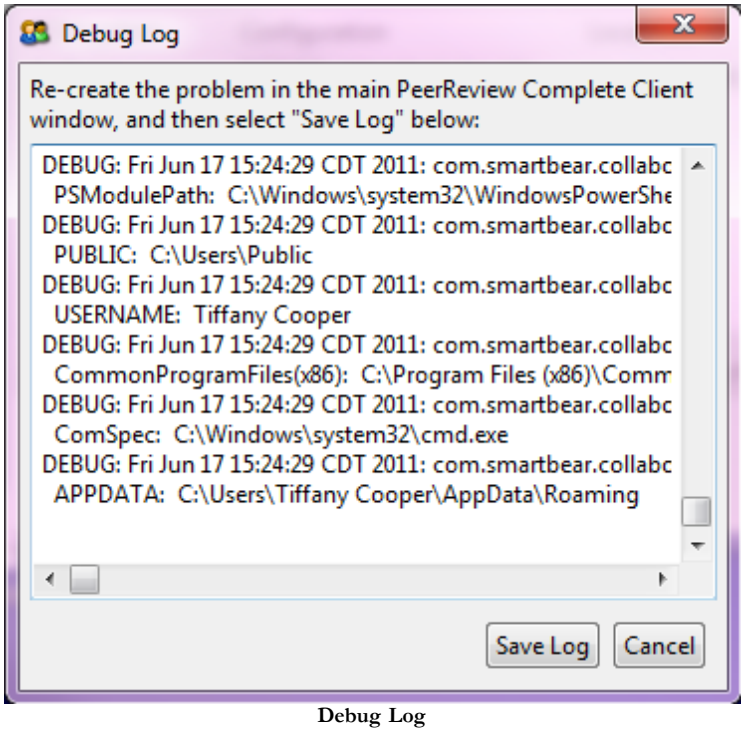


In the Miscellaneous tab, you can change the refresh interval (in seconds) and the preferred browser. Checking **Force New Browser Window** will force PeerReview Complete to open any new instances in a new window instead of a new tab in browsers that support such behavior.

© 2003-2011 SmartBear Software

4.3.4 Troubleshooting

Capturing a debugging log
If you are experiencing problems with the GUI Client, especially if the problem is related to your SCM system, it will help SmartBear technical support if you send in a debugging log. To capture a debugging log, select *Help -> Capture Debugging Log* from the system menu on the Main Screen 190.


Debug Log
Re-create the problem in the main PeerReview Complete Client window, and then select "Save Log" below:
DEBUG: Fri Jun 17 15:24:29 CDT 2011: com.smartbear.collabc
PSModulePath: C:\Windows\system32\WindowsPowerShe
DEBUG: Fri Jun 17 15:24:29 CDT 2011: com.smartbear.collabc
PUBLIC: C:\Users\Public
DEBUG: Fri Jun 17 15:24:29 CDT 2011: com.smartbear.collabc
USERNAME: Tiffany Cooper
DEBUG: Fri Jun 17 15:24:29 CDT 2011: com.smartbear.collabc
CommonProgramFiles(x86): C:\Program Files (x86)\Comm
DEBUG: Fri Jun 17 15:24:29 CDT 2011: com.smartbear.collabc
ComSpec: C:\Windows\system32\cmd.exe
DEBUG: Fri Jun 17 15:24:29 CDT 2011: com.smartbear.collabc
APPDATA: C:\Users\Tiffany Cooper\AppData\Roaming
Save Log
Cancel
Debug Log

After the **Debug Log** window has opened, go back to the Main Screen 190 window and recreate your problem. You should see log messages being created in the **Debug Log** window. After you have recreated the problem, go back to the **Debug Log** window and select the *Save Log* button. This will prompt you for a convenient place on your machine to save the log file.

Please send the log file along with a full description of what you were trying to do when the error occurred to support@smartbear.com.

4.4 Command Line Client

The Command Line Client is a cross-platform client interface to the PeerReview Complete server. It can be used by a human for uploading files, integrating with version control, and querying the server, or as a part of an automated script in a sophisticated ALM / build system.

This chapter includes:

- Configuration 197
- Commands 198
- Graphical Editor 200
- Uploading Diffs 200

© 2003-2011 SmartBear Software

- Troubleshooting 204
- Global Options Reference 204
- Command-line Reference 211

4.4.1 Configuration

The Command Line Client needs to be able to connect with your PeerReview Complete Server, connect with your SCM system, and may need to launch a graphical editor for selecting files to be uploaded to reviews. The following sections describe configuring 187 the Command Line Client for these actions.

Global Options

The Command Line Client supports many global options 204 and sub-command options. Global options may be specified on the command-line before the sub-command for each invocation, or they can be saved and applied to all future invocations using the set 215 sub-command.

Default settings for global options can also be set using one or more configuration files 198.

Configuring PeerReview Complete Server Connection

If you used the graphical installer 183, your connection to the PeerReview Complete Server should be configured already. Otherwise, you will be prompted when you try to connect. Try testing your configuration 197 to verify the configuration is detected correctly. You can also manually configure your connection 188 using the `login` 214 sub-command.

Configuring Version Control

In most cases, the Command Line Client can automatically detect your SCM system configuration. Try testing your configuration 197 to verify the configuration is detected correctly.

If the Command Line Client is unable to detect your SCM system, or if you want to override the detected settings, you can specify SCM configuration settings using global options 204:

- AccuRev 292
- ClearCase 329
- CMVC 318
- CVS 307
- MKS 361
- Perforce 375
- StarTeam 301
- Subversion 405
- Surround 422
- Team Foundation 352
- Vault 394

Testing Configuration Settings

© 2003-2011 SmartBear Software

To test the current configuration settings, go to the place on your local machine where your files are checked-out (if any), and execute:

`ccollab info` 213

This prints the current effective PeerReview Complete and version control configuration settings. If you have not specified the scm 205 global option, the Command Line Client will attempt to automatically detect your SCM configuration.

You will see some error messages if the configuration is not valid. If all goes well you should see something like this:

```
Connecting to PeerReview Complete server http://myserver:8080
Connected as: John Doe (jdoe)
Auto-detecting SCM System for 'C:\mycode'
Detected Subversion
SCM Username: jdoe
  SCM Config: repo=http://mysvnserver/repos/myrepo
```

Graphical Editor Configuration

The Command Line Client uses the default system text editor to display files to be uploaded for review. You can override this to point to any other text editor using the `editor` 205 global option.

Some editors on some platforms are launched as detached processes. This will cause the Command Line Client to continue before the editor is closed, losing any changes made to the file list. The `editor-prompt` 205 global option can be used to pause the Command Line Client after the editor is launched and wait for user keyboard input before continuing.

Configuration Files

PeerReview Complete uses several configuration files 188 to store default global options. When a user saves a global option 204 setting (using `ccollab set` 215), the setting is stored inside a directory called `.smartbear` inside the user's home directory. (Under Windows, the "home directory" is your "`Documents and Settings`" Profile directory.)

Global options specified by command-line switches 204 override settings stored in the configuration files.

4.4.2 Commands

Basic Commands

The Command Line Client provides a few basic commands which are unrelated to uploading files to the PeerReview Complete Server:

- `help` 213 - Display help on using the Command Line Client
- `login` 214 - Verify and/or change connection to the PeerReview Complete Server
- `info` 213 - Validates server connection and SCM configuration
- `set` 215 - Save a global option 204 setting

© 2003-2011 SmartBear Software

Upload Commands

The primary function of the Command Line Client is to upload review materials (files) to the PeerReview Complete Server. The recommended commands to use vary by Version Control System:

- AccuRev Commands 288
- ClearCase Commands 322
- CMVC Commands 318
- CVS Commands 302
- Git Commands 311
- Mercurial Integration 343
- MKS Commands 358
- Perforce Commands 366
- Rational Synergy Commands 336
- StarTeam Commands 297
- Subversion Commands 396
- Team Foundation Commands 349
- Vault Commands 391
- Visual SourceSafe Commands 356

In addition to the Version Control specific commands, the Command Line Client provides a few basic upload commands that work for any (or no) Version Control system:

- `addfiles` 217 - Upload local files without diffs
- `adddiffs` 219 - Upload local or arbitrary diffs (see Uploading Diffs 200)

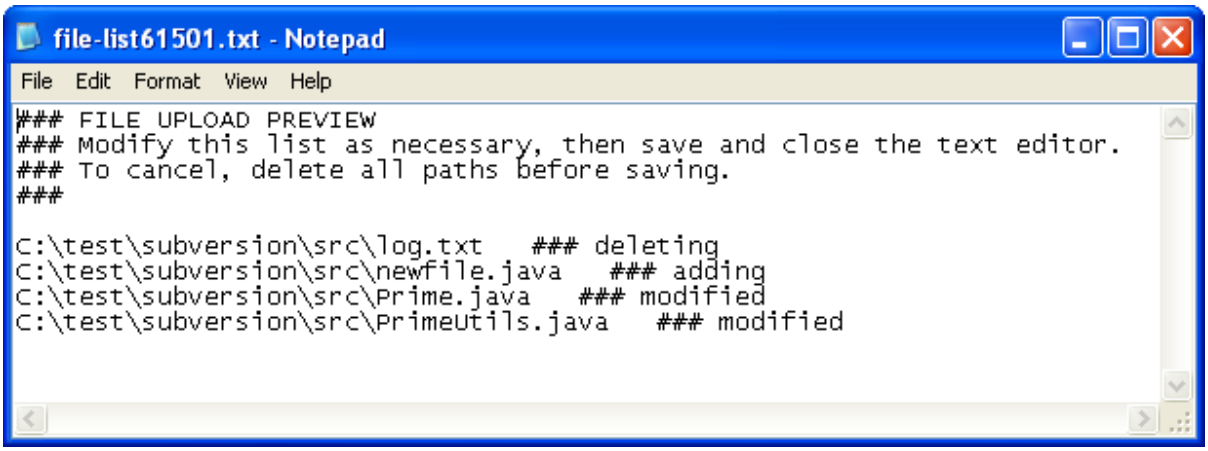
Scripting Commands

The Command Line Client also includes many commands meant for constructing custom scripts, usually used for integrating with external systems or implementing custom behaviors. For more information see Scripting 443.

© 2003-2011 SmartBear Software

4.4.3 Graphical Editor

The addchanges 216 and addactivity 229 commands can launch a graphical editor with a file list so you can review the files about to be uploaded and make any corrections you wish:


file-list61501.txt - Notepad
File Edit Format View Help
FILE UPLOAD PREVIEW
Modify this list as necessary, then save and close the text editor.
To cancel, delete all paths before saving.
###
C:\test\subversion\src\log.txt ### deleting
C:\test\subversion\src\newfile.java ### adding
C:\test\subversion\src\Prime.java ### modified
C:\test\subversion\src\PrimeUtils.java ### modified

The ### sequence is a comment. If all lines with files are deleted the operation is aborted. In this example there are two modified files, one file being added, and one being deleted.

If not otherwise configured, the command-line uses the default system editor to display the file list. See Graphical Editor Configuration 198 for information on specifying a different text editor to be used.

4.4.4 Uploading Diffs

You can upload arbitrary file differences using the `ccollab adddiffs` 219 command. This subject is complex enough to warrant its own section in the manual.

Uploading Local File Diffs

You can upload local file differences with the `adddiffs` command:

```
ccollab adddiffs <review> path-from path-to
```

The `path-from` and `path-to` fields are paths to two files or directories to compare, and <*review*> is an integer review-id, 'new', 'ask', or 'last'. For example, this command would upload any differences found as a change-set as a new review:

```
ccollab adddiffs new /old/directory /new/directory
```

Uploading Arbitrary Diffs

You can also upload arbitrary differences with the `adddiffs` command. Assuming textual diffs have already been generated in a file called `diffs.txt`, use this:

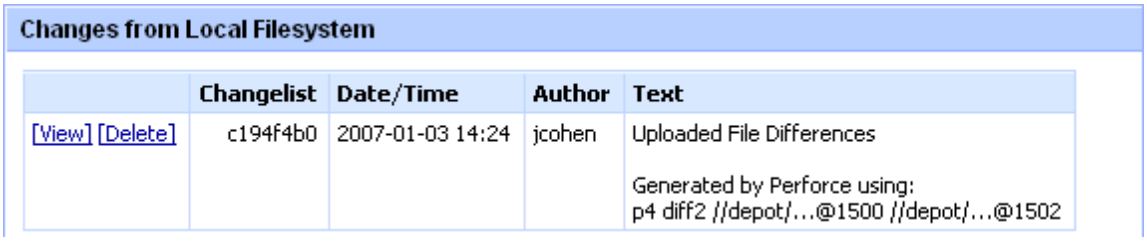
```
ccollab adddiffs review diffs.txt
```

© 2003-2011 SmartBear Software

Or you can pipe the results of another diff command right into the `ccollab` process using a single dash instead of a file path. For example, this command uses the GNU `diff` command to generate a unified diff format, then uploads it directly into review #2534:

```
diff -u100 dir1 dir2 | ccollab adddiffs 2534 -
```

After you upload diffs to the server you will see the "changelist" appear similar to this:



Changes from Local Filesystem

	Changelist	Date/Time	Author	Text
[View] [Delete]	c194f4b0	2007-01-03 14:24	jcohen	Uploaded File Differences Generated by Perforce using: p4 diff2 //depot/...@1500 //depot/...@1502

Uploading Differences from Version Control

If you want to upload differences generated by a version control command-line tool, you should use the `ccollab add*diffs` command specific to your version control system (if available) instead of `ccollab adddiffs`.

Version Control specific `add*diffs` commands:

- `addardiffs` 220 (AccuRev 288)
- `addcvsdiffs` 221 (CVS 302)
- `addgitdiffs` 222 (Git 311)
- `addhgdiffs` 223 (Mercurial 343)
- `addp4diffs` 224 (Perforce 366)
- `addstdiffs` 225 (StarTeam 297)
- `addsvndiffs` 220 (Subversion 396)
- `addvssdiffs` 226 (Visual SourceSafe 356)

These commands integrate with version control servers and supply exactly the right command-line switches to the underlying version control command-line to minimize server transmissions and maximize features like fully reproducing both previous and current versions of the document.

Many of the pitfalls with diffs described below are avoided when using `ccollab add*diffs`.

Supported Diff Formats

There are many kinds of diff formats out there. Our software supports many of them; if you find a format that isn't working, please let us know 24 so we can get your diff format into our unit tests.

© 2003-2011 SmartBear Software

- **Unified**
  The unified format starts with a two-line header naming the previous and name versions of a file, plus optional file information. Then it follows with diff-chunks starting with "@@" and then having both unmodified lines and added/removed lines. An exact definition of the format can be found here.
  Any number of these file-diffs can be strung together. Our software parses all of the files and uploads each as a "before" and "after" version. Context lines are preserved; lines skipped are reconstructed (more about this later 203).

- **UltraCompare Text Format**
  We support the proprietary text format emitted by the 3rd party commercial command-line tool UltraCompare.
  This is a Unicode diff with full file context making it an excellent format that preserves all file content.
  Although this utility emits only one file difference at a time, you can concatenate any number of these diffs together and upload them in one shot through the `ccollab` command-line. This can be especially handy if you're writing a script to automate this process.

- **Subversion Style**
  Subversion outputs a two-line header for each file; one starts with "Index:" and names the file in question, the next is a row of equal signs. Following the header is any of three types of diff output; we support all three but you should use the unified format for best results 203. Any number of these file-diffs can be strung together.
  Note that with Subversion you should use `addsvndiffs` 220 instead of `adddiffs`.

- **Perforce Style**
  Perforce has a variety of output formats; we support all of them. One is just the Unified format specified above. The other uses a one-line file header that shows the "before" and "after" file paths and version numbers with other file data, and then uses any of three types of diff output to encode the diffs. We support all three but you should use the unified format for best results 203. Any number of these file-diffs can be strung together.
  Note that with Perforce you should use `addp4diffs` 224 instead of `adddiffs`.

- **CVS Style**
  Similar to Subversion Style (see above) but has additional lines of output besides just those two file header markers.
  Note that with CVS you should use `addcvsdiffs` 221 instead of `adddiffs`.

- **GNU Format**
  The GNU format shows just additions, deletions, and modifications with no file context. An exact definition of the format can be found here.
  We don't recommend you use GNU format because (1) it doesn't support file names or multiple files and (2) it doesn't support context lines which causes problems when reconstructing files 203.

- **RCS Format**
  The RCS format shows just additions and deletions with no file context.
  We don't recommend you use RCS format because (1) it doesn't support file names or multiple files and (2) it doesn't support context lines which causes problems when reconstructing files 203.

© 2003-2011 SmartBear Software

- **StarTeam Format**
  StarTeam has a proprietary diff format for text files that provides complete lines of context for the revisions compared.
  Note that with StarTeam you should use `addstdiffs` 225 instead of `adddiffs`.

- **Araxis Merge, BeyondCompare, GuiffyMerge, GNU diff, xdiff, etc..**
  With these diff utilities, use an export option that creates unified diffs with lots of context. See the guidelines below 203.

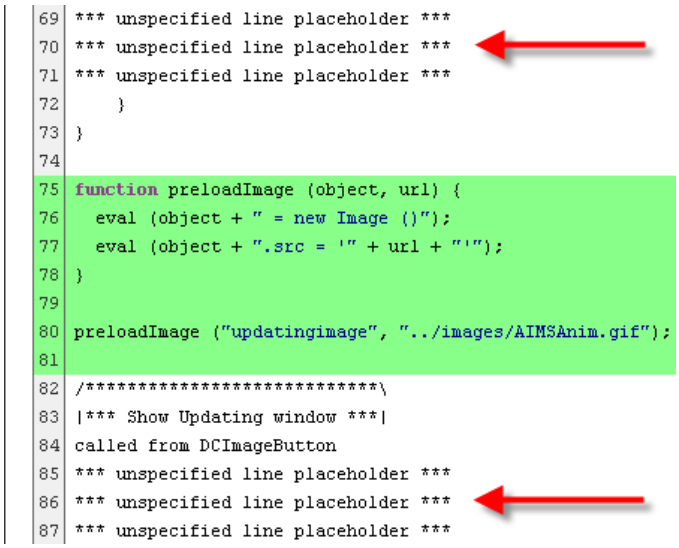
When the command-line utility is uploading diffs it will print out the format that was detected. If it complains that the format is unknown, attach a copy of your diff data to an email to our tech support department 24 so we can add support for your format.

Context Lines and Reconstructing Files

Most diff formats skip over lines that were unchanged, possibly leaving a few lines of "context" around each modification. This might be OK for "patch" utilities but it's not good when you want to do a review. Why not?

When you do the review, if you don't have many lines of context you can't see much beyond the diffs themselves. Our software is smart enough to reconstruct the two files as well as possible, retaining all line numbers, but of course there will be gaps.

Here's how the side-by-side view 160 looks with some differences that only had three lines of context:


69 *** unspecified line placeholder ***
70 *** unspecified line placeholder ***
71 *** unspecified line placeholder ***
72 }
73 }
74
75 function preloadImage (object, url) {
76 eval (object + " = new Image ()");
77 eval (object + ".src = '" + url + "'");
78 }
79
80 preloadImage ("updatingimage", "../images/AIMSAnim.gif");
81
82 /*****************************\
83 |*** Show Updating window ***|
84 called from DCImageButton
85 *** unspecified line placeholder ***
86 *** unspecified line placeholder ***
87 *** unspecified line placeholder ***

You can see the "`*** unspecified line placeholder ***`" text where the diff skipped over lines.

How to prevent this? It depends on how you're generating diffs. Here are some guidelines:

- Always use Unified diff format 201, not RCS or GNU. Those others don't support lines of context *at all*, nor do they support filenames so your uploads will be "anonymous."

© 2003-2011 SmartBear Software

- Always include a lot of context with your Unified diff format. The way to specify this varies depending on your diff utility. For example, using GNU `diff` the command-line option is `-u1000` where `"1000"` is the number of lines of context. Selecting a big number like 1000 is not unreasonable! Reviewers will thank you for it.
- Use the version-control specific `ccollab add*diffs` 201 commands whenever you can. This automatically selects the right command-line switches to maximize context while minimizing data transmissions.

4.4.5 Troubleshooting

Check here for common problems and their solutions.

Known Issues
Check the Known Issues Appendix 460 to see whether SmartBear already knows about this issue.

Version History
Check the Version History on the SmartBear website to see if this issue has been resolved in a later release.

I try to run `ccollab` but it says File Not Found.
Under Windows, sometimes the installer is not able to update the `PATH` variable due to system configuration or permissions. Or the `PATH` variable is updated but the system requires a log-out or log-in before it's in the `PATH`.

Under Unix systems, the `PATH` variable needs to be updated if you didn't install in a standard location such as `/bin` or `/usr/bin`.

Running with extra logging turned on
Running the command-line client with the --debug 205 option will create a log file called `ccollab.log` in your current working directory. If you contact tech support with a problem and they cannot reproduce it, this is usually what they'll tell you to do.

4.4.6 Global Options Reference

Usage:

```
ccollab [global-options] command [command-options]
```

Global Options can be specified on the command-line, or they can be saved and applied to every command using 'ccollab set 215'.

General Options

© 2003-2011 SmartBear Software

Option	Description
--browser <value>	Browser to launch (e.g. 'firefox', 'ie', or 'default')
--debug	Run in debug mode, create a ccollab.log file in current directory
--editor <value>	External editor to use for editing file list
--editor-prompt	Prompt user for input to signal editing of file list complete
--force-new-browser	Force browser to open new window (only works for some platforms and browsers)
--no-browser	Do not pop up a web browser when the command completes
--non-interactive	Don't interact with user
--password <value>	Password to use when connecting to the application server
--quiet	Don't display progress messages
--scm <value>	Manually sets the SCM system type, e.g. "perforce" or "none" for no SCM System. Clearing this option causes the SCM system to be auto-detected
--server-proxy-host <value>	Proxy server URL to use to connect to the application server
--server-proxy-port <value>	Proxy server port number to use to connect to the application server
--url <value>	URL to use when connecting to the application server
--user <value>	Username to use when connecting to the application server

AccuRev-specific Options

© 2003-2011 SmartBear Software

Option	Description
--accurev-depot <value>	AccuRev depot name
--accurev-exe <value>	Full path to the `accurev` command-line client
--accurev-anc-algorithm <value>	Which algorithm to use when calculating the predecessor, either predecessor, previous-occupant, or basis-version

ClearCase-specific Options

Option	Description
--clearcase-update-snapshot	Whether to update ClearCase snapshot views prior to uploading files for review

CVS-specific Options

Option	Description
--cvs-exe <value>	Full path to the `cvs` command-line client
--cvsroot <value>	Connection to the CVS repository

Git-specific Options

Option	Description
--git-exe <value>	Full path to the 'git' command line client

Mercurial-specific Options

© 2003-2011 SmartBear Software

Option	Description
--mercurial-exe <value>	Full path to the 'mercurial (hg)' command line client

MKS Source-specific Options

Option	Description
--mks-host <value>	MKS Source server name
--mks-port <value>	MKS Source server port
--mks-user <value>	MKS Source user name
--mks-passwd <value>	MKS Source user password
--mks-expand-keywords	Whether to expand keywords in MKS source files

Perforce-specific Options

Option	Description
--p4 <value>	Full path to the P4 executable
--p4port <value>	How to connect to the Perforce server
--p4user <value>	Perforce user name
--p4passwd <value>	Perforce password or ticket
--p4client <value>	Mapping of Perforce server data to the local machine
--p4-ignore-integration-history <value>	Ignore integration history when calculating predecessor
--p4charset <value>	Perforce character set used for translation of Unicode files

© 2003-2011 SmartBear Software

Option	Description
--p4-require-empty-default-changelist	If true, don't allow uploads if the default changelist contains files
--p4-specify-command-charset <value>	Should a character set be specified for communication with Perforce

Rational Synergy-specific Options

Option	Description
--ccm-exe <value>	Full path to the `ccm` command-line executable
--ccm-user <value>	User Name to use when starting a Rational Synergy session
--ccm-passwd <value>	Password to use when starting a Rational Synergy session
--ccm-engine-host <value>	Host the Rational Synergy Engine will run on
--ccm-database-path <value>	Path of the Rational Synergy database to connect with
--ccm-local-database-path <value>	Path of the local Rational Synergy database, typically c:/temp/ccm or /tmp/ccm
--ccm-remote-client	Start Rational Synergy sessions as a Remote Client

Rational Team Concert-specific Options

Option	Description

StarTeam-specific Options

© 2003-2011 SmartBear Software

Option	Description
--stcmd-exe <value>	Full path to the StarTeam `stcmd` command-line client

Subversion-specific Options

Option	Description
--svn-exe <value>	Full path to the `svn` command-line executable
--svn-look-exe <value>	Full path to the `svnlook` command-line executable (used by Subversion triggers)
--svn-repo-url <value>	Subversion repository URL
--svn-user <value>	Subversion user name
--svn-passwd <value>	Subversion password
--svn-require-client-certificate-password <value>	Use this if you have a non-empty and unsaved password for your SSL Client Certificate
--svn-auto-add	Treat unversioned files as if they had been added to Subversion
--svn-recurse-externals	Recurse in the 'svn:externals' directories as if they were part of the same repository
--svn-repo-path <value>	Full path to the repository (used by Subversion Triggers)

Surround SCM-specific Options

Option	Description
--surround-exe <value>	Full path to the Surround SCM `sscm` command-line client
--surround-server <value>	Surround SCM server address (server:port)

© 2003-2011 SmartBear Software

Option	Description
--surround-single-sign-on	Surround SCM single sign-on
--surround-user <value>	Surround SCM user name
--surround-passwd <value>	Surround SCM password

Team Foundation Server-specific Options

Option	Description
--tfs-collection <value>	Team Foundation collection name (2010 and later only)
--tfs-server <value>	Team Foundation server name (2005 and 2008)
--tfs-exe <value>	Full path to the Team Foundation `tf` command-line client
--tfs-user <value>	Team Foundation alternate user name (atypical)
--tfs-passwd <value>	Team Foundation alternate user name password

Vault-specific Options

Option	Description
--vault-profile <value>	Vault profile name
--vault-server <value>	Vault server url (n/a with profile)
--vault-repo <value>	Vault repository (n/a with profile)
--vault-user <value>	Vault user (n/a with profile)
--vault-passwd <value>	Vault password (n/a with profile)
--vault-proxy-domain <value>	Vault proxy domain
--vault-proxy-server <value>	Vault proxy server

© 2003-2011 SmartBear Software

Option	Description
--vault-proxy-port <value>	Vault proxy port
--vault-proxy-user <value>	Vault proxy user
--vault-proxy-passwd <value>	Vault proxy password

Visual SourceSafe-specific Options

Option	Description
--vss-passwd <value>	SourceSafe password
--vss-exe <value>	Full path to the SourceSafe `ss` command-line client
--vss-user <value>	SourceSafe user name

4.4.7 Command-line Reference

```
Usage: ccollab [global-options] command [command-options]
```

`ccollab` - PeerReview Complete Command-line Client

Sub-Commands

Sub-Command	Description
help 213	Display help on using the Command Line Client
info 213	Validates server connection and SCM configuration
login 214	Verify and/or change connection to the server
logout 214	Clears the stored password and/or login ticket id, and invalidates the user's login ticket on the server
set 215	Save a global option setting

© 2003-2011 SmartBear Software

Sub-Command	Description
addchanges 216	Attaches locally-modified files to a review
addfiles 217	Attaches local files to a review without diffs
addchangelist 218	Attaches an atomic changelist to a review
adddiffs 219	Attaches file differences to a Review
addsvndiffs 220	Uploads diffs generated from the svn diff command
addardiffs 220	Uploads diffs generated from accurev diff command
addcvsdiffs 221	Uploads diffs generated from cvs diff command
addgitdiffs 222	Uploads diffs generated by git diff command
addhgdiffs 223	Uploads diffs generated by hg diff command
addp4diffs 224	Uploads diffs generated from p4 diff2 command
addstdiffs 225	Uploads diffs generated from the stcmd diff command
addvssdiffs 226	Uploads diffs generated from 'ss diff' command
addversions 227	Attaches any 2 given versions to a review
addactivity 229	Attaches file versions in a ClearCase activity to a review
actionitems 230	List current action items
addtrack 230	Attaches file parts for a Track associated with the given defect and release to a review
addstream 231	Attaches pending differences from an AccuRev stream
browse 232	Launch a browser to the PeerReview Complete Server homepage
commit 232	Commit changes in the review
addp4job 233	Adds all numbered local changes that fix a job to the review

© 2003-2011 SmartBear Software

Sub-Command	Description
addurls 233	Attaches urls to a review
admin 234	Perform administration tasks

4.4.7.1 help

```
Usage: ccollab [global-options] help [--show-all] [<topic>]
```

`help` - Display help on using the Command Line Client

Command Options

Option	Required?	Description
--show-all	N	Display all commands, including those not recommended for the current SCM system
<topic>	N	A specific command to display help for, or 'global-options' to display help on global options

If the topic is not specified, displays help on all commands.

Examples:

ccollab help

ccollab help global-options

ccollab help addchanges

ccollab help admin review-xml

ccollab help batch-xml

4.4.7.2 info

```
Usage: ccollab [global-options] info
```

© 2003-2011 SmartBear Software

`info` - Validates server connection and SCM configuration

Uses the configured global options to validate the connection to the PeerReview Complete Server.

Also validates the SCM configuration. If the 'scm' global option is not set, tries to automatically detect the SCM configuration for the current working directory.

4.4.7.3 login

```
Usage: ccollab [global-options] login [<url>] [<user>] [<password>]
[<server-proxy-host>]  [<server-proxy-port>]
```

`login` - Verify and/or change connection to the server

Command Options

Option	Required?	Description
<url>	N	New URL to server
<user>	N	New username
<password>	N	New password
<server-proxy-host>	N	New proxy server URL
<server-proxy-port>	N	New proxy server port

If no arguments are specified, this command will first try to use the existing configuration to connect to the PeerReview Complete Server.

If the url, user, or password arguments are not specified, the user will be prompted to supply the necessary configuration information.

If the new configuration settings result in a valid connection to a PeerReview Complete Server, they will be saved to the configuration file.

4.4.7.4 logout

```
Usage: ccollab [global-options] logout
```

© 2003-2011 SmartBear Software

`logout` - Clears the stored password and/or login ticket id, and invalidates the user's login ticket on the server

4.4.7.5 set

```
Usage: ccollab [global-options] set [<key>] [<value>]
```

`set` - Save a global option setting

Command Options

Option	Required?	Description
<key>	N	Key to save
<value>	N	Value to save

Save a global option setting in a config file, so it will be set upon subsequent invocations of 'ccollab'. To clear a global option setting, set it to "". Global options saved by 'ccollab set' are overridden by options specified on the command-line.

With the exception of 'password', any global option can be saved using this command. For the list of global options run 'ccollab help global-options'.

Examples:

ccollab set url http://collab.mycompany.com

ccollab set user myusername

If the <value> argument is not supplied on the command-line, 'ccollab set' will prompt for it.

If no arguments are supplied on the command-line, 'ccollab set' will print the current values of all the global options.

Note that the user should run 'ccollab login' in order to save the global option 'password'.

© 2003-2011 SmartBear Software

4.4.7.6 addchanges

```
Usage: ccollab [global-options] addchanges [--diffbranch <value> --upload-
comment <value>] <review> <file-spec> [<file-spec> ...]
```

`addchanges` - Attaches locally-modified files to a review

Command Options

Option	Required?	Description
--diffbranch <value>	N	(ClearCase only) specify either branch-name or "predecessor" to determine the earlier version for upload
--upload-comment <value>	N	Comment used to upload files (default is "Local changes")
<review>	Y	Must be either an integer review-id, 'new', 'ask', or 'last'
<file-spec> [<file-spec> ...]	Y	Local files or directories to scan for modified files

(See Version Control specific description - AccuRev 294 ClearCase 330 CVS 308 Git 316 Mercurial 347 Subversion 407 Surround SCM 423 Team Foundation Server 353 MKS Source 362 Vault 395 Uploaded Files)

Uploads locally modified files controlled by an SCM to the PeerReview Complete server.

The first argument is the review specifier, then any number of files or directories can be given. Files are selected as-is; directories are scanned recursively for locally-modified files, which might include edits, additions, deletions, branches, integrations, moves, copies, and so on.

After the scan is complete, a file list is presented in a graphical editor so you can review the files about to be uploaded and make any corrections you wish.

For example, to create a new review and add all changes in the current directory and below, plus the file foo.txt, you would use:

© 2003-2011 SmartBear Software

ccollab addchanges new . foo.txt

To upload modified files from the current working directory and all subdirectories to review 123:

ccollab addchanges 123 .

To upload file foo.txt and modified files from c:\dev\project into a brand new review:

ccollab addchanges new foo.txt c:\dev\project

4.4.7.7 addfiles

```
Usage: ccollab [global-options] addfiles [--relative-to <value> --upload-
comment <value>] <review> <file-path> [<file-path> ...]
```

`addfiles` - Attaches local files to a review without diffs

Command Options

Option	Required?	Description
--relative-to <value>	N	Absolute parent path to calculate file-paths relative to if no SCM system
--upload-comment <value>	N	Comment used to upload files (default is "Local changes")
<review>	Y	Must be either an integer review-id, 'new', 'ask', or 'last'
<file-path> [<file-path> ...]	Y	Local files to upload

Uploads local files to the PeerReview Complete server. Files are uploaded without a "previous version" so the entire file will be shown without differences.

If you upload the same file path on more than one occasion, the server will start making diffs of each version against the previous.

© 2003-2011 SmartBear Software

The server displays the full file system path of the files, unless you specify the --relative-to option.

It is an error to specify a directory or a file that does not exist.

For example, to upload files foo.txt and c:\dev\bar.txt into a brand new review:

```
ccollab addfiles new foo.txt c:\dev\bar.txt
```

To upload file c:\dev\bar.txt into review #62:

```
ccollab addfiles 62 c:\dev\bar.txt
```

4.4.7.8 addchangelist

```
Usage: ccollab [global-options] addchangelist <review> <changelist>
[<changelist>  ...]
```

`addchangelist` - Attaches an atomic changelist to a review

Command Options

Option	Required?	Description
<review>	Y	Must be either an integer review-id, 'new', 'ask', or 'last'
<changelist> [<changelist> ...]	Y	SCM-specific ID of an atomic set of changes

(See Version Control specific description - Perforce 376 AccuRev 295 Subversion 408 Team Foundation Server 354 ClearCase 331 CMVC 319 MKS Source 363 Surround SCM 424 Rational Synergy 340 Rational Team Concert)

Uploads changelists from an SCM to the PeerReview Complete server.

The first argument is the review specifier, subsequent arguments are the IDs of the changelists to upload.

© 2003-2011 SmartBear Software

For example, to upload changelist 4321 and 7568 to a new review:

```
ccollab addchangelist new 4321 7568
```

For example, to upload changelist 5432 and 12654 to review 111:

```
ccollab addchangelist 111 5432 12654
```

4.4.7.9 adddiffs

```
Usage: ccollab [global-options] adddiffs [--relative --upload-comment
<value>] <review> <before> [<after>]
```

`adddiffs` - Attaches file differences to a Review

Command Options

Option	Required?	Description
--relative	N	Upload as paths relative to the before or after directory
--upload-comment <value>	N	Comment used to upload files
<review>	Y	Must be either an integer review-id, 'new', 'ask', or 'last'
<before>	Y	If `after` is also given, a directory of files to treat as base files or file to use as base file (before changes were made). If no `after` argument is given, this is a file containing multi-file textual diffs, or '-' to read diffs from stdin
<after>	N	Directory of files to scan for differences or file to use as modified file(after changes have been made)

© 2003-2011 SmartBear Software

4.4.7.10 addsvndiffs

```
Usage: ccollab [global-options] addsvndiffs [--upload-comment <value>]
<review> [<user-diff-arg> [<user-diff-arg> ...]]
```

`addsvndiffs` - Uploads diffs generated from the svn diff command

Command Options

Option	Required?	Description
--upload-comment <value>	N	Comment used to upload files (defaults to command-line arguments)
<review>	Y	Must be either an integer review-id, 'new', 'ask', or 'last'
<user-diff-arg> [<user-diff-arg> ...]	N	Command-line arguments to pass directly to the diff command

Upload differences between arbitrary versions of files in Subversion using 'svn diff'.

Do not use diff arguments that affect the diff output such as --diff-cmd, -x [--extensions], --summarize, or --non-interactive. The PeerReview Complete command-line client will automatically select an output format that ensures you will get all the data you need on the server.

Examples:

ccollab addsvndiffs 698 -r 2:16

ccollab addsvndiffs new -r PREV http://my.svn.server/svn/repo

ccollab addsvndiffs 698 -c 10534 src/java

ccollab addsvndiffs new OLDURL[@OLDREV] NEWURL[@NEWREV]

4.4.7.11 addardiffs

```
Usage: ccollab [global-options] addardiffs [--upload-comment <value>]
<review> [<user-diff-arg> [<user-diff-arg> ...]]
```

© 2003-2011 SmartBear Software

`addardiffs` - Uploads diffs generated from accurev diff command

Command Options

Option	Required?	Description
--upload-comment <value>	N	Comment used to upload files (defaults to command-line arguments)
<review>	Y	Must be either an integer review-id, 'new', 'ask', or 'last'
<user-diff-arg> [<user-diff-arg> ...]	N	Command-line arguments to pass directly to the diff command

Upload differences between arbitrary versions of files in AccuRev using 'accurev diff'.

Do not use diff arguments that affect the diff output such as -c or -G. The PeerReview Complete command-line client will automatically select an output format that ensures you will get all the data you need on the server.

Examples:

ccollab addardiffs 698 -a -b -v "My Stream"

4.4.7.12 addcvsdiffs

```
Usage: ccollab [global-options] addcvsdiffs [--upload-comment <value>]
<review> [<user-diff-arg> [<user-diff-arg> ...]]
```

`addcvsdiffs` - Uploads diffs generated from cvs diff command

Command Options

Option	Required?	Description
--upload-comment <value>	N	Comment used to upload files (defaults to command-line arguments)

© 2003-2011 SmartBear Software

Option	Required?	Description
<review>	Y	Must be either an integer review-id, 'new', 'ask', or 'last'
<user-diff-arg> [<user-diff-arg> ...]	N	Command-line arguments to pass directly to the diff command

Upload differences between arbitrary versions of files in CVS using 'cvs diff'.

Do not use diff arguments that affect the diff output such as -N --new-file, -u -U --unified, -c -C --context, --brief, -e -ed, -f --forward-ed, -n --rcs, or -y --side-by-side. The PeerReview Complete command-line client will automatically select an output format that ensures you will get all the data you need on the server.

For example, to upload all changes between labels build-391 and build-401:

```
ccollab addcvsdiffs review -r build-391 -r build-401
```

To upload all changes between dates 2006-01-01 and 2006-02-01:

```
ccollab addcvsdiffs review -D 2006-01-01 -D 2006-02-01
```

To upload all changes in the last 7 days:

```
ccollab addcvsdiffs review -D "7 days ago"
```

4.4.7.13 addgitdiffs

```
Usage: ccollab [global-options] addgitdiffs [--upload-comment <value>]
<review> [<user-diff-arg> [<user-diff-arg> ...]]
```

`addgitdiffs` - Uploads diffs generated by git diff command

Command Options

© 2003-2011 SmartBear Software

Option	Required?	Description
--upload-comment <value>	N	Comment used to upload files (defaults to command-line arguments)
<review>	Y	Must be either an integer review-id, 'new', 'ask', or 'last'
<user-diff-arg> [<user-diff-arg> ...]	N	Command-line arguments to pass directly to the diff command

Upload differences between arbitrary versions of files in Git using 'git diff'.

Do not use diff arguments that affect the diff output such as -U unified. The PeerReview Complete command-line client will automatically select an output format that ensures you will get all the data you need on the server.

For example, to upload all changes between the revision 8 revisions ago and the revision 4 revisions ago:

ccollab addgitdiffs review master~8 master~4

To upload all changes in your local working directory:

ccollab addgitdiffs review

4.4.7.14 addhgdiffs

```
Usage: ccollab [global-options] addhgdiffs [--upload-comment <value>]
<review> [<user-diff-arg> [<user-diff-arg> ...]]
```

`addhgdiffs` - Uploads diffs generated by hg diff command

Command Options

© 2003-2011 SmartBear Software

Option	Required?	Description
--upload-comment <value>	N	Comment used to upload files (defaults to command-line arguments)
<review>	Y	Must be either an integer review-id, 'new', 'ask', or 'last'
<user-diff-arg> [<user-diff-arg> ...]	N	Command-line arguments to pass directly to the diff command

Upload differences between arbitrary versions of files in Mercurial using 'hg diff'.

Do not use diff arguments that affect the diff output such as -U unified. The PeerReview Complete command-line client will automatically select an output format that ensures you will get all the data you need on the server.

For example, to upload all changes between revisions 4 and 8:

```
ccollab addhgdiffs review -r 4 -r 8
```

To upload all changes in your local working directory:

```
ccollab addhgdiffs review .
```

4.4.7.15 addp4diffs

```
Usage: ccollab [global-options] addp4diffs [--upload-comment <value>]
<review> [<p4-diff-arg> [<p4-diff-arg> ...]]
```

`addp4diffs` - Uploads diffs generated from p4 diff2 command

Command Options

© 2003-2011 SmartBear Software

Option	Required?	Description
--upload-comment <value>	N	Comment used to upload files (defaults to command-line arguments)
<review>	Y	Must be either an integer review-id, 'new', 'ask', or 'last'
<p4-diff-arg> [<p4-diff-arg> ...]	N	Options which should be passed to the p4 diff2 command

Upload differences between arbitrary versions of files in Perforce using 'p4 diff2'.

Do not use diff arguments that affect the diff output such as -q, -t, -d<flag>, or -s<flag>. The PeerReview Complete command-line client will automatically select an output format that ensures you will get all the data you need on the server.

Examples:

ccollab addp4diffs 698 //depot/file1 //depot/file2

ccollab addp4diffs new //depot/...@1523 //depot/...

ccollab addp4diffs 698 -b my-branch

4.4.7.16 addstdiffs

```
Usage: ccollab [global-options] addstdiffs [--upload-comment <value>]
<review> [<user-diff-arg> [<user-diff-arg> ...]]
```

`addstdiffs` - Uploads diffs generated from the stcmd diff command

Command Options

Option	Required?	Description
--upload-comment <value>	N	Comment used to upload files (defaults to command-line arguments)
<review>	Y	Must be either an integer review-id, 'new', 'ask', or 'last'

© 2003-2011 SmartBear Software

Option	Required?	Description
<user-diff-arg> [<user-diff-arg> ...]	N	Command-line arguments to pass directly to the diff command

Upload differences between arbitrary versions of files in StarTeam using 'stcmd diff'.

Do not use diff arguments that affect the diff output such as -c or -e. The PeerReview Complete command-line client will automatically select an output format that ensures you will get all the data you need on the server.

Examples:

ccollab addstdiffs 698 -p "project" -vl LABEL1 -vl LABEL2

ccollab addstdiffs new -p "project" -vd "2007-01-01" *.cpp

4.4.7.17 addvssdiffs

```
Usage: ccollab [global-options] addvssdiffs [--upload-comment <value>]
<review> [<user-diff-arg> [<user-diff-arg> ...]]
```

`addvssdiffs` - Uploads diffs generated from 'ss diff' command

Command Options

Option	Required?	Description
--upload-comment <value>	N	Comment used to upload files (defaults to command-line arguments)
<review>	Y	Must be either an integer review-id, 'new', 'ask', or 'last'
<user-diff-arg> [<user-diff-arg> ...]	N	Command-line arguments to pass directly to the diff command

Upload differences between arbitrary versions of files in Visual SourceSafe using 'ss diff'.

© 2003-2011 SmartBear Software

Do not use diff arguments that affect the diff output such as -o, -b, -h, -?, -du<width> where width is less than 128, or -dx<context> where context is less than 10. The PeerReview Complete command-line client will automatically select an output format that ensures you will get all the data you need on the server.

Examples:

ccollab addvssdiffs 698 HelloWorld.java -yusername,password

ccollab addvssdiffs new HelloWorld.java -v2

ccollab addvssdiffs 698 HelloWorld.java -v4~1

ccollab addvssdiffs 698 $/Project/Status.java c:\project\Status.java

4.4.7.18 addversions

```
Usage: ccollab [global-options] addversions [--upload-comment <value>]
<review> [<version-spec> [<version-spec> ...]]
```

`addversions` - Attaches any 2 given versions to a review

Command Options

Option	Required?	Description
--upload-comment <value>	N	Comment used to upload files (default is "Local changes")
<review>	Y	Must be either an integer review-id, 'new', 'ask', or 'last'

© 2003-2011 SmartBear Software

Option	Required?	Description
<version-spec> [<version-spec> ...]	N	Versions to be reviewed are given by <path> <version> [<previous version>], where <path> is the filename or server path of the file, <version> is the version to be reviewed, and <previous version> is an optional version that <version> should be diff'ed against. If <previous version> is not specified, the predecessor version of <version> will be used. If no <version-spec> is given on the command line, then one or more version specs will be read from stdin, one <version-spec> per line as described above. Note that arguments containing spaces must be quoted to be parsed correctly

(See Version Control specific description - ClearCase 332 CMVC 320 MKS Source 365 Perforce 378 Surround SCM 425 Rational Synergy 342)

Uploads specific versions of a file or files to a review. A version-spec is given by a filename followed by a version number, and optionally, a previous version number:

file-name version-number [previous-version-number]

If no previous version number is explicity given, diffs will be generated with the predecessor version as determined from the SCM system.

Adding versions for a single file can be done with command line arguments given after the review number. To upload versions for multiple files, the file and versions may be submitted on lines read from standard input. If the path and versions are not given on the command line, the client will

© 2003-2011 SmartBear Software

expect to read them from stdin. Each line read should be in the format shown above. Some example command lines for reading versions from stdin:

```
ccollab addversions 86753
ccollab addversions last < versionlist.txt
cat versionlist.txt | ccollab addversions new
```

When supplying the version list in the command line client editor or an input file, the keyword 'local' can be used to denote the version corresponding to the local version of the file. The 'local' keyword can only be used for the first version argument, not the previous version.

4.4.7.19 addactivity

```
Usage: ccollab [global-options] addactivity [--diffintegration --upload-
comment <value>] <review> <activity-name> [<activity-name> ...]
```

`addactivity` - Attaches file versions in a ClearCase activity to a review

Command Options

Option	Required?	Description
--diffintegration	N	Use LATEST version from default integration stream, if available, as the predecessor version
--upload-comment <value>	N	Comment used to upload files (default is "Local changes")
<review>	Y	Must be either an integer review-id, 'new', 'ask', or 'last'
<activity-name> [<activity-name> ...]	Y	Specify one or more ClearCase activities by name, or use 'all' to include all activities in the current view

© 2003-2011 SmartBear Software

Attaches file versions from one or more activities to a review. For each file in the activity, the latest version in the activity and the predecessor of the earliest version in the activity are uploaded to the review. You can specify multiple activity names on the command line. You can also use the word 'all' as the activity name to include all activities in the current view. If multiple activities are given, the changeset uploaded is the union of all changes in each of the activities.

You can use the --diffintegration option to include the default integration stream in determining version content for review. For the most recent version, this option will scan the versions of each file in the given activity as well as the versions of those files in any associated rebase activities, to determine the most recent version. For the previous version for each file in the activity, this option will use the LATEST version in the default integration branch (if available) as the previous version.

4.4.7.20 actionitems

```
Usage: ccollab [global-options] actionitems
```

`actionitems` - List current action items

4.4.7.21 addtrack

```
Usage: ccollab [global-options] addtrack [--upload-comment <value>]
<review> <defect-name> <release-name> [<root-path>]
```

`addtrack` - Attaches file parts for a Track associated with the given defect and release to a review

Command Options

Option	Required?	Description
--upload-comment <value>	N	Comment used to upload files (default is "Local changes")
<review>	Y	Must be either an integer review-id, 'new', 'ask', or 'last'
<defect-name>	Y	Associated CMVC defect
<release-name>	Y	Associated CMVC release
<root-path>	N	Path to local source files (CMVC top directory)

© 2003-2011 SmartBear Software

Upload files that are Parts in a CMVC Track to a review. If the root path is not specified, file content is extracted from CMVC, comparing the latest version in the Track to its predecessor version. If the root path is specified, the root path is used to resolve the file parts of the Track to the local files on disk, and these are compared against the file versions in the Track.

For example, to upload files that are already checked into CMVC:

ccollab addtrack review defect release

If you have a local snapshot directory with local modifications and the CMVC files exist and are part of a Track:

ccollab addtrack review defect release rootpath

rootpath specifies the root of the local sandbox that the file path names in the Track are relative to.

4.4.7.22 addstream

```
Usage: ccollab [global-options] addstream [--upload-comment <value>]
<review> [<stream>]
```

`addstream` - Attaches pending differences from an AccuRev stream

Command Options

Option	Required?	Description
--upload-comment <value>	N	Comment used to upload files
<review>	Y	Must be either an integer review-id, 'new', 'ask', or 'last'
<stream>	N	Specify an AccuRev stream by name, or leave blank to use the current workspace

Attaches differences pending promotion in an AccuRev stream.

© 2003-2011 SmartBear Software

4.4.7.23 browse

```
Usage: ccollab [global-options] browse [--review <value>]
```

`browse` - Launch a browser to the PeerReview Complete Server homepage

Command Options

Option	Required?	Description
--review <value>	N	Must be either an integer review-id, 'ask', or 'last'

4.4.7.24 commit

```
Usage: ccollab [global-options] commit [--comment <value> --dismiss-only --
force] <review>
```

`commit` - Commit changes in the review

Command Options

Option	Required?	Description
--comment <value>	N	Comment for reviewed changes
--dismiss-only	N	Just dismiss the Action Item
--force	N	Ignore potential problems
<review>	Y	Must be either an integer review-id, 'ask', or 'last'

(See Version Control specific description - Perforce 379 Subversion 410 Team Foundation Server 355 ClearCase 336 CVS 311)

Commit the changes from a pre-commit review to sourcecontrol. Be sure to include a relevant comment.

© 2003-2011 SmartBear Software

4.4.7.25 addp4job

```
Usage: ccollab [global-options] addp4job <review> <job-name>
```

`addp4job` - Adds all numbered local changes that fix a job to the review

Command Options

Option	Required?	Description
<review>	Y	Must be either an integer review-id, 'new', 'ask', or 'last'
<job-name>	Y	Perforce changes that fix this job will be added to the review

4.4.7.26 addurls

```
Usage: ccollab [global-options] addurls [--upload-comment <value>] <review>
<urls> [<urls> ...]
```

`addurls` - Attaches urls to a review

Command Options

Option	Required?	Description
--upload-comment <value>	N	Comment used to attach URLs (default is "Attached URLs")
<review>	Y	Must be either an integer review-id, 'new', 'ask', or 'last'
<urls> [<urls> ...]	Y	URLs to attach

Adds URLs to a review in the PeerReview Complete server. The content of the URLs are not attached to the review and remain on the origin server. URLs are not reviewed with differences, so adding the same URL more than once will have no practical effect.

© 2003-2011 SmartBear Software

For example, to review the contents of a wiki page in a brand new review:

```
ccollab addurls new http://wiki/Topic_Page
```

To review the contents of a wiki page in review #62:

```
ccollab addurls 62 http://wiki/Topic_Page
```

4.4.7.27 admin

```
Usage: ccollab [global-options] admin sub-command [sub-command-options]
```

`admin` - Perform administration tasks

Sub-Commands

Sub-Command	Description
syncusers 235	Synchronizes the Perforce user list with the PeerReview Complete server
review-xml 235	Extract the state of a review in XML form
batch 236	Process multiple commands with one connection
wget 237	Get the contents of a URL on the server
config 238	Administer Server Configuration
review 239	Administer Reviews
user 256	Administer Users
trigger 262	SCM Triggers
group 272	Administer Groups
changelist 277	Administer Changelists

© 2003-2011 SmartBear Software

4.4.7.27.1 syncusers

```
Usage: ccollab [global-options] admin syncusers
```

`syncusers` - Synchronizes the Perforce user list with the PeerReview Complete server

Causes all users from Perforce to be mirrored into PeerReview Complete.

The algorithm is smart enough not to overwrite existing users, only adding new users, so this can be run periodically by a script to keep the lists in sync.

This is not necessary -- or desirable -- if you are using LDAP authentication.

You must be logged in as an administrator to execute this command.

4.4.7.27.2 review-xml

```
Usage: ccollab [global-options] admin review-xml [--xpath <value> --xsl-
file <value>] <review> [<review> ...]
```

`review-xml` - Extract the state of a review in XML form

Command Options

Option	Required?	Description
--xpath <value>	N	XPath expression to evaluate against XML
--xsl-file <value>	N	XSL file to use to post-process XML, or '-' to read XSL from STDIN
<review> [<review> ...]	Y	Reviews to dump to XML

You may specify an XPath expression to evaluate against the resulting XML.

Examples:

© 2003-2011 SmartBear Software

Get ID of 'last' review:

ccollab admin review-xml last --xpath string(//reviews/review/@reviewId)

Get title of review 1354:

ccollab admin review-xml 1354 --xpath //reviews/review/general/title/text()

Get number of defects in review 2478 with 'Major' severity:

ccollab admin review-xml 2478 --xpath count(//reviews/review/defects/defect[custom-defect-fields/severity='Major'])

You may alternately specify an XSL file to post-process the XML.

4.4.7.27.3 batch

```
Usage: ccollab [global-options] admin batch <commands-file>
```

`batch` - Process multiple commands with one connection

Command Options

Option	Required?	Description
<commands-file>	Y	File containing commands in XML format, or "-" to read from STDIN

Running multiple commands in batch form ensures that the 'last' review and defect variables stay consistent, even if there are multiple concurrent invocations of the client. It's also easier to encode non-ascii characters in XML.

Global Options specified in the XML override the current Global Options.

Command options should be given without the leading double-dash. Both option and argument values are given as text values inside elements. For option and argument elements that can accept multiple values, specify a new element for each value.

© 2003-2011 SmartBear Software

Example XML file:

```
<batch-commands>
        <global-options>
                <no-browser/>
                <non-interactive/>
        </global-options>
        <addchanges>
                <upload-comment>These are my latest changes!</upload-comment>
                <review>new</review>
                <file-spec>.</file-spec>
        </addchanges>
        <addfiles>
                <review>last</review>
                <file-path>checklist.txt</file-path>
                <file-path>foo/bar.java</file-path>
                <file-path>biz/buzz.java</file-path>
        </addfiles>
        <browse>
                <review>last</review>
        </browse>
</batch-commands>
```

Run 'ccollab help batch-xml' for the full XML Schema accepted by this command.

4.4.7.27.4 wget

```
Usage: ccollab [global-options] admin wget <relative-url>
```

`wget` - Get the contents of a URL on the server

Command Options

© 2003-2011 SmartBear Software

Option	Required?	Description
<relative-url>	Y	URL relative to the server

Gets the contents of a URL on the PeerReview Complete server, taking care of authentication. Note this command will effectively cause a license to be consumed if the URL requires a licensed user.

4.4.7.27.5 config

```
Usage: ccollab [global-options] admin config sub-command [sub-command-
options]
```

`config` - Administer Server Configuration

Sub-Commands

Sub-Command	Description
custom-field 238	Administer Custom Fields

4.4.7.27.5.1 custom-field

```
Usage: ccollab [global-options] admin config custom-field sub-command [sub-
command-options]
```

`custom-field` - Administer Custom Fields

Sub-Commands

Sub-Command	Description
edit 238	Edit an existing custom field definition

```
Usage: ccollab [global-options] admin config custom-field edit [--drop-
down-item <value> [<value> ...]] <custom-field>
```

© 2003-2011 SmartBear Software

`edit` - Edit an existing custom field definition

Command Options

Option	Required?	Description
--drop-down-item <value> [<value> ...]	N	Drop-down items for a 'Drop-down List' Type custom field
<custom-field>	Y	ID of the custom field to edit

Custom field ID's are displayed in the administration screens in the PeerReview Complete web interface.

4.4.7.27.6 review

```
Usage: ccollab [global-options] admin review sub-command [sub-command-
options]
```

`review` - Administer Reviews

Sub-Commands

Sub-Command	Description
create 240	Create a new Review
edit 241	Edit an existing Review
delete 242	Delete an existing Review
cancel 242	Cancel an in progress Review
wait 243	Wait for a response in a Review
finish 243	Finish current phase of a Review
copy-participants 244	Copy Participants from one Review to another
set-participants 244	Set Participants for a Review
participant 245	Administer Review Participants
comment 247	Administer Review Comments

© 2003-2011 SmartBear Software

Sub-Command	Description
defect 249	Administer Review Defects
conversation 253	Administer Review Conversations

4.4.7.27.6.1 create

```
Usage: ccollab [global-options] admin review create [--creator <value> --
custom-field <value> [<value> ...] --deadline <value> --display-
changelists-as <value> --group <value> --participant-custom-field <value>
[<value> ...] --restrict-access <value> --template <value> --title
<value>]
```

`create` - Create a new Review

Command Options

Option	Required?	Description
--creator <value>	N	The login of the creator of the Review, defaults to logged-in user if not specified
--custom-field <value> [<value> ...]	N	Setting for custom fields, in the form field-name=field-value (e.g. "Overview=this is the overview")
--deadline <value>	N	Deadline in the format 'yyyy-MM-dd'
--display-changelists-as <value>	N	Default to showing changelists rolled up('single') or individually ('distinct')
--group <value>	N	Guid of the group to specify
--participant-custom-field <value> [<value> ...]	N	Setting for participant custom fields, in the form field-name=field-value
--restrict-access <value>	N	Restrict Access (anyone/group/participants/group-and-participants)

© 2003-2011 SmartBear Software

Option	Required?	Description
--template <value>	N	
--title <value>	N	

4.4.7.27.6.2 edit

```
Usage: ccollab [global-options] admin review edit [--custom-field <value>
[<value> ...] --deadline <value> --display-changelists-as <value> --group
<value> --participant-custom-field <value> [<value> ...] --restrict-access
<value> --template <value> --title <value>] <review>
```

`edit` - Edit an existing Review

Command Options

Option	Required?	Description
--custom-field <value> [<value> ...]	N	Setting for custom fields, in the form field-name=field-value (e.g. "Overview=this is the overview")
--deadline <value>	N	Deadline in the format 'yyyy-MM-dd'
--display-changelists-as <value>	N	Default to showing changelists rolled up('single') or individually ('distinct')
--group <value>	N	Guid of the group to specify
--participant-custom-field <value> [<value> ...]	N	Setting for participant custom fields, in the form field-name=field-value
--restrict-access <value>	N	Restrict Access (anyone/group/participants/group-and-participants)
--template <value>	N	
--title <value>	N	

© 2003-2011 SmartBear Software

Option	Required?	Description
<review>	Y	Must be either an integer review-id, 'ask', or 'last'

Note when setting custom fields you have to specify the review first.

For example:

ccollab admin review edit "last" --custom-field "Overview=this is the overview"

4.4.7.27.6.3 delete

```
Usage: ccollab [global-options] admin review delete <review>
```

`delete` - Delete an existing Review

Command Options

Option	Required?	Description
<review>	Y	Must be either an integer review-id, 'ask', or 'last'

Reviews can only be deleted in the planning phase.

4.4.7.27.6.4 cancel

```
Usage: ccollab [global-options] admin review cancel <review>
```

`cancel` - Cancel an in progress Review

Command Options

Option	Required?	Description
<review>	Y	Must be either an integer review-id, 'ask', or 'last'

© 2003-2011 SmartBear Software

4.4.7.27.6.5 wait

```
Usage: ccollab [global-options] admin review wait [--until <value>]
<review>
```

`wait` - Wait for a response in a Review

Command Options

Option	Required?	Description
--until <value>	N	Activity to wait until ('Any', 'Author', 'File', or 'Poke')
<review>	Y	Must be either an integer review-id, 'ask', or 'last'

Indicate that you are waiting until the specified action. You will be notified when the action occurs. Defaults to any type of activity.

4.4.7.27.6.6 finish

```
Usage: ccollab [global-options] admin review finish [--until <value>]
<review>
```

`finish` - Finish current phase of a Review

Command Options

Option	Required?	Description
--until <value>	N	Activity to finish until ('Any', 'Author', 'File', or 'Poke')
<review>	Y	Must be either an integer review-id, 'ask', or 'last'

Finish the current phase of the Review for the logged in user.

© 2003-2011 SmartBear Software

For example:

If the Review is in the 'Planning' phase and the logged in user is the Creator of the Review, this command will move the Review to the 'Inspection' phase.

For a Review in the 'Inspection' phase where the logged in user is a Reviewer in that Review, if the Review has outstanding defects this command will cause the Review to move in to the 'Rework' phase. If the Review has no defects then the Review will be 'Completed' (unless there are other Reviewers who have not finished yet).

4.4.7.27.6.7 copy-participants

```
Usage: ccollab [global-options] admin review copy-participants <from-
review> <to-review>
```

`copy-participants` - Copy Participants from one Review to another

Command Options

Option	Required?	Description
<from-review>	Y	Review to copy Participants from. Must be either an integer review-id, 'ask', or 'last'
<to-review>	Y	Review to copy Participants to. Must be either an integer review-id, 'new', 'ask', or 'last'

The Participants specified in 'from-review' are added to 'to-review' in the same Roles. Existing Participants in the 'to' Review are not removed.

For example, to create a new Review and initialize it with Participants used in the last Review, run 'ccollab admin review copy-participants last new'

4.4.7.27.6.8 set-participants

```
Usage: ccollab [global-options] admin review set-participants [--
participant <value> [<value> ...]] <review>
```

© 2003-2011 SmartBear Software

`set-participants` - Set Participants for a Review

Command Options

Option	Required?	Description
--participant <value> [<value> ...]	Y	Participants, in the form 'role=user'
<review>	Y	Must be either an integer review-id, 'ask', or 'last'

Participants are added or removed as necessary to make the Review match the options of this command.

Example:

ccollab admin review set-participants 12345 --participant author=roy --participant reviewer=brandon

4.4.7.27.6.9 participant

```
Usage: ccollab [global-options] admin review participant sub-command [sub-command-options]
```

`participant` - Administer Review Participants

Sub-Commands

Sub-Command	Description
assign 245	Assign a User to a participant Role in a Review
remove 246	Remove a participant from a Review
poke 246	Poke a participant in a Review

```
Usage: ccollab [global-options] admin review participant assign <review> <participant> <role>
```

© 2003-2011 SmartBear Software

`assign` - Assign a User to a participant Role in a Review

Command Options

Option	Required?	Description
<review>	Y	Must be either an integer review-id, 'ask', or 'last'
<participant>	Y	Username of participant
<role>	Y	Role of participant in Review

```
Usage: ccollab [global-options] admin review participant remove <review>
<participant>
```

`remove` - Remove a participant from a Review

Command Options

Option	Required?	Description
<review>	Y	Must be either an integer review-id, 'ask', or 'last'
<participant>	Y	Username of participant

```
Usage: ccollab [global-options] admin review participant poke <review>
<participant>
```

`poke` - Poke a participant in a Review

Command Options

© 2003-2011 SmartBear Software

Option	Required?	Description
<review>	Y	Must be either an integer review-id, 'ask', or 'last'
<participant>	Y	Username of participant

4.4.7.27.6.10 comment

```
Usage: ccollab [global-options] admin review comment sub-command [sub-
command-options]
```

`comment` - Administer Review Comments

Sub-Commands

Sub-Command	Description
create 247	Create a Comment on a Review
redact 249	Redact a Comment on a Review

```
Usage: ccollab [global-options] admin review comment create [--annotation
--conversation-id <value> --label <value> --line-number <value> --page
<value> --file <value> --version <value> --x <value> --y <value>] <review>
<comment>
```

`create` - Create a Comment on a Review

Command Options

Option	Required?	Description
--annotation	N	File annotation (overwrites any existing annotation)
--conversation-id <value>	N	ID of the first Comment in an existing Conversation

© 2003-2011 SmartBear Software

Option	Required?	Description
--label <value>	N	Label
--line-number <value>	N	Line number in file
--page <value>	N	Page number
--file <value>	N	Path of file in Review
--version <value>	N	Version ID of file in Review
--x <value>	N	X coordinate
--y <value>	N	Y coordinate
<review>	Y	Must be either an integer review-id, 'ask', or 'last'
<comment>	Y	

You can specify various options to control the location where the comment is created.

Annotation: --annotation

Overall: (default - no extra options necessary)

Label: --label <label>

Line: --line-number <number>

Coordinate: --page <number> --x <number> --y <number>

To specify an existing Conversation in a file, it's best to specify --conversation-id <id> to disambiguate situations where there's more than one Conversation at a particular location.

Only one of 'file' or 'version' may be specified.

Example:

ccollab admin review comment create 123 "overall review comment"

ccollab admin review comment create 345 --file "foobar.c" --annotation "ignore this file"

ccollab admin review comment create 567 --version 789 --line-number 36 "I like this line"

© 2003-2011 SmartBear Software

```
Usage: ccollab [global-options] admin review comment redact <comment-id>
```

`redact` - Redact a Comment on a Review

Command Options

Option	Required?	Description
<comment-id>	Y	Comment ID to redact

Mark a Comment 'Redacted'. Note the ID of a Comment can be obtained from 'ccollab admin review-xml'.

4.4.7.27.6.11 defect

```
Usage: ccollab [global-options] admin review defect sub-command [sub-command-options]
```

`defect` - Administer Review Defects

Sub-Commands

Sub-Command	Description
create 249	Create a Defect
edit 251	Edit a Defect
mark-fixed 251	Mark a Defect as 'Fixed'
mark-external 252	Mark a Defect as 'Tracked Externally'
mark-not-external 252	Mark a Defect as not 'Tracked Externally' (Open)
mark-open 252	Mark a Defect as 'Open'
delete 253	Delete a Defect

© 2003-2011 SmartBear Software

```
Usage: ccollab [global-options] admin review defect create [--conversation-
id <value> --custom-field <value> [<value> ...] --label <value> --line-
number <value> --page <value> --file <value> --version <value> --x <value>
--y <value>] <review> <comment>
```

`create` - Create a Defect

Command Options

Option	Required?	Description
--conversation-id <value>	N	ID of the first Comment in an existing Conversation
--custom-field <value> [<value> ...]	N	Setting for custom fields, in the form field-name=field-value (e.g. "Severity=Major")
--label <value>	N	Label
--line-number <value>	N	Line number in file
--page <value>	N	Page number
--file <value>	N	Path of file in Review
--version <value>	N	Version ID of file in Review
--x <value>	N	X coordinate
--y <value>	N	Y coordinate
<review>	Y	Must be either an integer review-id, 'ask', or 'last'
<comment>	Y	

You can specify various options to control the location where the defect is created.

Overall: (default - no extra options necessary)

Label: --label <label>

Line: --line-number <number>

Coordinate: --page <number> --x <number> --y <number>

© 2003-2011 SmartBear Software

To specify an existing Conversation in a file, it's best to specify --conversation-id <id> to disambiguate situations where there's more than one Conversation at a particular location.

Only one of 'file' or 'version' may be specified.

Example:

ccollab admin review defect create 123 "overall review defect"

ccollab admin review defect create 567 --file "bizbaz.java" "overall file defect"

ccollab admin review defect create 567 --version 789 --line-number 36 "defect on a line"

```
Usage: ccollab [global-options] admin review defect edit [--comment <value>
--custom-field <value> [<value> ...]] <defect>
```

`edit` - Edit a Defect

Command Options

Option	Required?	Description
--comment <value>	N	
--custom-field <value> [<value> ...]	N	Setting for custom fields, in the form field-name=field-value (e.g. "Severity=Major")
<defect>	Y	Must be either an integer defect-id or 'last'

```
Usage: ccollab [global-options] admin review defect mark-fixed <defect>
```

`mark-fixed` - Mark a Defect as 'Fixed'

Command Options

© 2003-2011 SmartBear Software

Option	Required?	Description
<defect>	Y	Must be either an integer defect-id or 'last'

```
Usage: ccollab [global-options] admin review defect mark-external <defect>
<external-name>
```

`mark-external` - Mark a Defect as 'Tracked Externally'

Command Options

Option	Required?	Description
<defect>	Y	Must be either an integer defect-id or 'last'
<external-name>	Y	External name to refer to externally tracked Defect

```
Usage: ccollab [global-options] admin review defect mark-not-external
<defect>
```

`mark-not-external` - Mark a Defect as not 'Tracked Externally' (Open)

Command Options

Option	Required?	Description
<defect>	Y	Must be either an integer defect-id or 'last'

```
Usage: ccollab [global-options] admin review defect mark-open <defect>
```

© 2003-2011 SmartBear Software

`mark-open` - Mark a Defect as 'Open'

Command Options

Option	Required?	Description
<defect>	Y	Must be either an integer defect-id or 'last'

```
Usage: ccollab [global-options] admin review defect delete <defect>
```

`delete` - Delete a Defect

Command Options

Option	Required?	Description
<defect>	Y	Must be either an integer defect-id or 'last'

4.4.7.27.6.12 conversation

```
Usage: ccollab [global-options] admin review conversation sub-command [sub-command-options]
```

`conversation` - Administer Review Conversations

Sub-Commands

Sub-Command	Description
mark-read 254	Mark a conversation on a Review as 'read'
accept 255	Mark a conversation on a Review as 'accepted'

© 2003-2011 SmartBear Software

```
Usage: ccollab [global-options] admin review conversation mark-read [--
conversation-id <value> --label <value> --line-number <value> --page
<value> --file <value> --version <value> --x <value> --y <value>] <review>
```

`mark-read` - Mark a conversation on a Review as 'read'

Command Options

Option	Required?	Description
--conversation-id <value>	N	ID of the first Comment in an existing Conversation
--label <value>	N	Label
--line-number <value>	N	Line number in file
--page <value>	N	Page number
--file <value>	N	Path of file in Review
--version <value>	N	Version ID of file in Review
--x <value>	N	X coordinate
--y <value>	N	Y coordinate
<review>	Y	Must be either an integer review-id, 'ask', or 'last'

You can specify various options to identify the conversation that should be marked read.

Overall: (default - no extra options necessary)

Label: --label <label>

Line: --line-number <number>

Coordinate: --page <number> --x <number> --y <number>

To specify an existing Conversation in a file, it's best to specify --conversation-id <id> to disambiguate situations where there's more than one Conversation at a particular location.

© 2003-2011 SmartBear Software

Only one of 'file' or 'version' may be specified.

Example:

ccollab admin review conversation mark-read 123

ccollab admin review conversation mark-read 567 --version 789 --line-number 36

ccollab admin review conversation mark-read 567 --file "bizbaz.java"

```
Usage: ccollab [global-options] admin review conversation accept [--
conversation-id <value> --label <value> --line-number <value> --page
<value> --file <value> --version <value> --x <value> --y <value>] <review>
```

`accept` - Mark a conversation on a Review as 'accepted'

Command Options

Option	Required?	Description
--conversation-id <value>	N	ID of the first Comment in an existing Conversation
--label <value>	N	Label
--line-number <value>	N	Line number in file
--page <value>	N	Page number
--file <value>	N	Path of file in Review
--version <value>	N	Version ID of file in Review
--x <value>	N	X coordinate
--y <value>	N	Y coordinate
<review>	Y	Must be either an integer review-id, 'ask', or 'last'

You can specify various options to identify the conversation that should be marked accepted.

Overall: (default - no extra options necessary)

© 2003-2011 SmartBear Software

Label: --label <label>

Line: --line-number <number>

Coordinate: --page <number> --x <number> --y <number>

To specify an existing Conversation in a file, it's best to specify --conversation-id <id> to disambiguate situations where there's more than one Conversation at a particular location.

Only one of 'file' or 'version' may be specified.

Example:

ccollab admin review conversation accept 123

ccollab admin review conversation accept 567 --version 789 --line-number 36

ccollab admin review conversation accept 567 --file "bizbaz.java"

4.4.7.27.7 user

```
Usage: ccollab [global-options] admin user sub-command [sub-command-
options]
```

`user` - Administer Users

Sub-Commands

Sub-Command	Description
create 256	Create a new User
edit 257	Edit an existing User
author-subscription 258	Administer Author Subscriptions
file-subscription 260	Administer File Subscriptions

4.4.7.27.7.1 create

```
Usage: ccollab [global-options] admin user create [--admin <value> --email
<value> --enabled <value> --full-name <value> --new-login <value> --
password <value> --phone-number <value>] <user>
```

© 2003-2011 SmartBear Software

`create` - Create a new User

Command Options

Option	Required?	Description
--admin <value>	N	Administrator status (y/n/yes/no/true/false/on/off)
--email <value>	N	Email address
--enabled <value>	N	Enablement status (y/n/yes/no/true/false/on/off)
--full-name <value>	N	Full name (enclose value in quotes if full name includes spaces)
--new-login <value>	N	New login / username for the user
--password <value>	N	Password
--phone-number <value>	N	Phone number
<user>	Y	Login or username of the user

Create a new user. You must be an Administrator to execute this command.

Examples:

ccollab admin user create alice

ccollab admin user create bob --phone-number 867-5309

4.4.7.27.7.2 edit

```
Usage: ccollab [global-options] admin user edit [--admin <value> --email
<value> --enabled <value> --full-name <value> --new-login <value> --
password <value> --phone-number <value>] <user>
```

`edit` - Edit an existing User

© 2003-2011 SmartBear Software

Command Options

Option	Required?	Description
--admin <value>	N	Administrator status (y/n/yes/no/true/false/on/off)
--email <value>	N	Email address
--enabled <value>	N	Enablement status (y/n/yes/no/true/false/on/off)
--full-name <value>	N	Full name (enclose value in quotes if full name includes spaces)
--new-login <value>	N	New login / username for the user
--password <value>	N	Password
--phone-number <value>	N	Phone number
<user>	Y	Login or username of the user

Edit the information of an existing user. You must be an Administrator to disable users or edit the information of another user.

4.4.7.27.7.3 author-subscription

```
Usage: ccollab [global-options] admin user author-subscription sub-command
[sub-command-options]
```

`author-subscription` - Administer Author Subscriptions

Sub-Commands

Sub-Command	Description
create 259	Create a Subscription to a Review Author
edit 259	Edit a Subscription to a Review Author
delete 259	Delete a Subscription to a Review Author

© 2003-2011 SmartBear Software

Author-based subscriptions allow a User to be automatically included in reviews where a given User is an Author. If no authors are selected for a given review, the Subscription will use the name of the Review creator instead.

```
Usage: ccollab [global-options] admin user author-subscription create
<user> <author> <role>
```

`create` - Create a Subscription to a Review Author

Command Options

Option	Required?	Description
<user>	Y	Login or username of the user
<author>	Y	Login or username of the Author
<role>	Y	Role in Review

```
Usage: ccollab [global-options] admin user author-subscription edit <user>
<author> <role>
```

`edit` - Edit a Subscription to a Review Author

Command Options

Option	Required?	Description
<user>	Y	Login or username of the user
<author>	Y	Login or username of the Author
<role>	Y	Role in Review

```
Usage: ccollab [global-options] admin user author-subscription delete
<user> <author>
```

© 2003-2011 SmartBear Software

`delete` - Delete a Subscription to a Review Author

Command Options

Option	Required?	Description
<user>	Y	Login or username of the user
<author>	Y	Login or username of the Author

4.4.7.27.7.4 file-subscription

```
Usage: ccollab [global-options] admin user file-subscription sub-command
[sub-command-options]
```

`file-subscription` - Administer File Subscriptions

Sub-Commands

Sub-Command	Description
create 260	Create a Subscription to a file Pattern
edit 261	Edit a Subscription to a file pattern
delete 261	Delete a Subscription to a file pattern

File pattern subscriptions allow you to be automatically included in reviews where particular files are under review. Select an Ant-style expression as the pattern to match the file(s) of interest. Ant-style expressions can include "*" to match any substring within a given directory, or "**" to match any substring of any length, ignoring directory separators.

If possible, avoid beginning the pattern with wildcard characters as it can significantly decrease the performance of subscription processing.

```
Usage: ccollab [global-options] admin user file-subscription create <user>
<file-pattern> <role>
```

`create` - Create a Subscription to a file Pattern

© 2003-2011 SmartBear Software

Command Options

Option	Required?	Description
<user>	Y	Login or username of the user
<file-pattern>	Y	File pattern to match
<role>	Y	Role in Review

```
Usage: ccollab [global-options] admin user file-subscription edit [--new-
file-pattern <value> --role <value>] <user> <file-pattern>
```

`edit` - Edit a Subscription to a file pattern

Command Options

Option	Required?	Description
--new-file-pattern <value>	N	New file pattern to match
--role <value>	N	Role in Review
<user>	Y	Login or username of the user
<file-pattern>	Y	File pattern to match

```
Usage: ccollab [global-options] admin user file-subscription delete <user>
<file-pattern>
```

`delete` - Delete a Subscription to a file pattern

Command Options

© 2003-2011 SmartBear Software

Option	Required?	Description
<user>	Y	Login or username of the user
<file-pattern>	Y	File pattern to match

4.4.7.27.8 trigger

```
Usage: ccollab [global-options] admin trigger sub-command [sub-command-
options]
```

`trigger` - SCM Triggers

Sub-Commands

Sub-Command	Description
ensure-review-started 262	Changelist cannot be submitted until review of this changelist exists
ensure-reviewed 264	Changelist cannot be submitted until review of this changelist has been completed
ensure-content-reviewed 266	Changelist cannot be submitted until review completed & file list/content is unchanged
ensure-diffs-reviewed 268	Ensure the reviewed diffs are the same as the diffs in the change
update-changelist 268	Automatically update Perforce changelist with information about the review of that changelist
create-review 270	Creates a new Review for a changelist if no Review already exists

4.4.7.27.8.1 ensure-review-started

```
Usage: ccollab [global-options] admin trigger ensure-review-started [--
ignore-integrate --review-id-regex <value>] <changelist-id>
```

`ensure-review-started` - Changelist cannot be submitted until review of this changelist exists

© 2003-2011 SmartBear Software

Command Options

Option	Required?	Description
--ignore-integrate	N	Allow integration changes to proceed without review
--review-id-regex <value>	N	Regular Expression that identifies Review ID in commit comment
<changelist-id>	Y	Changelist ID

(See Version Control specific description - Perforce 384 Subversion 411)

This trigger ensures that a Review has been started before changes can be committed to version control.

If a Review has not been started, the commit will fail and the user will get an error message explaining that the changes need to be reviewed before commit.

For Perforce, to install this trigger edit your Perforce Triggers list (see Perforce manual for details) and add a line like the following (except supply your installation directory for the ccollab application, replace <values> as appropriate, and put everything on one line even though the following text is word-wrapping):

ccollab submit //depot/... "/usr/bin/ccollab --url <collabUrl> --user <collabUser> --password <collabPasswd> --scm perforce --p4port <p4port> --p4user <p4user> --p4passwd <p4passwd> admin trigger ensure-review-started %changelist%"

On Unix systems the Perforce trigger system incorrectly splits up arguments by whitespace ignoring single quotes and other standard ways of indicating that data with spaces is actually a single argument. The solution is to use a separate bash script to call our trigger application.

For Subversion, to install this trigger you will need to create a pre-commit hook. If you already have a pre-commit hook, you can add our tool wherever it is appropriate; otherwise you will need to create an executable hook as described in the Subversion documentation (typically a batch file under Windows or a shell script under Linux/Mac).

Example Windows batch file:

© 2003-2011 SmartBear Software

"C:\Program Files\PeerReview Complete Client\ccollab.exe" --url <collabUrl> --user <collabUser> --password <collabPasswd> --scm subversion --svn-repo-path %1 --svn-look-exe "C:\Program Files\Subversion\bin\svnlook.exe" admin trigger ensure-review-started --review-id-regex "review:\s+(\d+)" %2 || exit 1

Example Linux/OSX shell script:

/collab/install/ccollab --url <collabUrl> --user <collabUser> --password <collabPasswd> --scm subversion --svn-repo-path $1 --svn-look-exe /usr/bin/svnlook admin trigger ensure-review-started --review-id-regex "review:\s+(\d+)" $2 || exit 1

Note our use of "exit 1" to ensure that the script terminates with a non-zero exit code if our trigger application fails.

For Perforce, you must supply credentials for a Perforce user that has at least read-only access.

For Perforce, changelists that result from creating a new branch are ignored. Creating a new branch does not involve a code change, so it does not require review.

For Perforce, integrating changes from one branch into another branch does make code changes and can introduce bugs, so by default this type of change is handled by the trigger. However, because integration changes frequently involve a very large number of files and because often it is necessary to integrate changes to many branches, this would require cumbersome reviews to be performed many times.

For this reason, the '--ignore-integrate' option can be used to ignore changelists consisting of only integration changes. In this case, enforcement of review happens externally to the system: for example, a review could be performed on one branch, but other branches could be just directly integrated.

For Subversion, you must specify the --svn-repo-path and --svn-look-exe global options.

For Subversion, you must require developers to put the review ID somewhere in the Subversion commit message. The format of this text is completely up to you; you must supply a Java-style regular expression that identifies this text and specifically calls out the review ID inside that text using the --review-id-regex command option.

4.4.7.27.8.2 ensure-reviewed

```
Usage: ccollab [global-options] admin trigger ensure-reviewed [--ignore-
integrate --review-id-regex <value>] <changelist-id>
```

© 2003-2011 SmartBear Software

`ensure-reviewed` - Changelist cannot be submitted until review of this changelist has been completed

Command Options

Option	Required?	Description
--ignore-integrate	N	Allow integration changes to proceed without review
--review-id-regex <value>	N	Regular Expression that identifies Review ID in commit comment
<changelist-id>	Y	Changelist ID

(See Version Control specific description - Perforce 385 Subversion 413)

This trigger ensures that a Review has been completed before changes can be committed to version control.

If a Review has not been completed, the commit will fail and the user will get an error message explaining that the changes need to be reviewed before commit.

For Perforce, to install this trigger edit your Perforce Triggers list (see Perforce manual for details) and add a line like the following (except supply your installation directory for the ccollab application, replace <values> as appropriate, and put everything on one line even though the following text is word-wrapping):

ccollab submit //depot/... "/usr/bin/ccollab --url <collabUrl> --user <collabUser> --password <collabPasswd> --scm perforce --p4port <p4port> --p4user <p4user> --p4passwd <p4passwd> admin trigger ensure-reviewed %changelist%"

On Unix systems the Perforce trigger system incorrectly splits up arguments by whitespace ignoring single quotes and other standard ways of indicating that data with spaces is actually a single argument. The solution is to use a separate bash script to call our trigger application.

For Subversion, to install this trigger you will need to create a pre-commit hook. If you already have a pre-commit hook, you can add our tool wherever it is appropriate; otherwise you will need to create an executable hook as described in the Subversion documentation (typically a batch file under Windows or a shell script under Linux/Mac).

© 2003-2011 SmartBear Software

Example Windows batch file:

"C:\Program Files\PeerReview Complete Client\ccollab.exe" --url <collabUrl> --user <collabUser> --password <collabPasswd> --scm subversion --svn-repo-path %1 --svn-look-exe "C:\Program Files\Subversion\bin\svnlook.exe" admin trigger ensure-reviewed --review-id-regex "review:\s+(\d+)" %2 || exit 1

Example Linux/OSX shell script:

/collab/install/ccollab --url <collabUrl> --user <collabUser> --password <collabPasswd> --scm subversion --svn-repo-path $1 --svn-look-exe /usr/bin/svnlook admin trigger ensure-reviewed --review-id-regex "review:\s+(\d+)" $2 || exit 1

Note our use of "exit 1" to ensure that the script terminates with a non-zero exit code if our trigger application fails.

For Perforce, you must supply credentials for a Perforce user that has at least read-only access.

For Perforce, changelists that result from creating a new branch are ignored. Creating a new branch does not involve a code change, so it does not require review.

For Perforce, integrating changes from one branch into another branch does make code changes and can introduce bugs, so by default this type of change is handled by the trigger. However, because integration changes frequently involve a very large number of files and because often it is necessary to integrate changes to many branches, this would require cumbersome reviews to be performed many times.

For this reason, the '--ignore-integrate' option can be used to ignore changelists consisting of only integration changes. In this case, enforcement of review happens externally to the system: for example, a review could be performed on one branch, but other branches could be just directly integrated.

For Subversion, you must specify the --svn-repo-path and --svn-look-exe global options.

For Subversion, you must require developers to put the review ID somewhere in the Subversion commit message. The format of this text is completely up to you; you must supply a Java-style regular expression that identifies this text and specifically calls out the review ID inside that text using the --review-id-regex command option.

4.4.7.27.8.3 ensure-content-reviewed

```
Usage: ccollab [global-options] admin trigger ensure-content-reviewed [--
ignore-integrate --no-keywords --review-id-regex <value>] <changelist-id>
```

© 2003-2011 SmartBear Software

`ensure-content-reviewed` - Changelist cannot be submitted until review completed & file list/ content is unchanged

Command Options

Option	Required?	Description
--ignore-integrate	N	Allow integration changes to proceed without review
--no-keywords	N	Disallow keyword expansion
--review-id-regex <value>	N	Regular Expression that identifies Review ID in commit comment
<changelist-id>	Y	Changelist ID

This trigger ensures that a Review has been completed and the file list and file content is unchanged before changes can be submitted to version control.

If a Review has not been completed, or the Review contains different files or file contents, the submit will fail and the user will get an error message explaining that the changes need to be reviewed before submit.

For Perforce, to install this trigger edit your Perforce Triggers list (see Perforce manual for details) and add a line like the following (except supply your installation directory for the ccollab application, replace <values> as appropriate, and put everything on one line even though the following text is word-wrapping):

ccollab content //depot/... "/usr/bin/ccollab --url <collabUrl> --user <collabUser> --password <collabPasswd> --scm perforce --p4port <p4port> --p4user <p4user> --p4passwd <p4passwd> admin trigger ensure-content-reviewed %changelist%"

On Unix systems the Perforce trigger system incorrectly splits up arguments by whitespace ignoring single quotes and other standard ways of indicating that data with spaces is actually a single argument. The solution is to use a separate bash script to call our trigger application.

For Perforce, you must supply credentials for a Perforce user that has at least read-only access.

© 2003-2011 SmartBear Software

For Perforce, changelists that result from creating a new branch are ignored. Creating a new branch does not involve a code change, so it does not require review.

For Perforce, integrating changes from one branch into another branch does make code changes and can introduce bugs, so by default this type of change is handled by the trigger. However, because integration changes frequently involve a very large number of files and because often it is necessary to integrate changes to many branches, this would require cumbersome reviews to be performed many times.

For this reason, the '--ignore-integrate' option can be used to ignore changelists consisting of only integration changes. In this case, enforcement of review happens externally to the system: for example, a review could be performed on one branch, but other branches could be just directly integrated.

4.4.7.27.8.4 ensure-diffs-reviewed

```
Usage: ccollab [global-options] admin trigger ensure-diffs-reviewed [--
ignore-integrate --no-keywords --review-id-regex <value>] <changelist-id>
```

`ensure-diffs-reviewed` - Ensure the reviewed diffs are the same as the diffs in the change

Command Options

Option	Required?	Description
--ignore-integrate	N	Allow integration changes to proceed without review
--no-keywords	N	Disallow keyword expansion
--review-id-regex <value>	N	Regular Expression that identifies Review ID in commit comment
<changelist-id>	Y	Changelist ID

Ensure that the difference between the current content and proposed content of each file is the same as the difference between the base and proposed content in the review by comparing the line-wise changes. com.smartbear.scm.impl.perforce.PerforceSystem@406199

4.4.7.27.8.5 update-changelist

```
Usage: ccollab [global-options] admin trigger update-changelist [--
description-prefix <value> --description-template <value>] <changelist-
spec>
```

© 2003-2011 SmartBear Software

`update-changelist` - Automatically update Perforce changelist with information about the review of that changelist

Command Options

Option	Required?	Description
--description-prefix <value>	Y	Prefix text for review line in changelist description
--description-template <value>	Y	Template text for review line in changelist description
<changelist-spec>	Y	%changelist% for 'commit' trigger, or %formfile% for 'form-in change' trigger

If you are reviewing changelists prior to submit, you might want some text in your changelist description saying the changelist was reviewed, which review ID, participants, and so on.

Running this command on the 'submit' trigger causes a description of the review to be added or updated in the changelist text when a pending changelist is submitted. If you also run this trigger on 'form-in change' then users can see the data update whenever they edit the changelist.

To install this trigger edit your Perforce Triggers list (see Perforce manual for details) and add a line like the following (except supply your installation directory for the ccollab application, replace <values> as appropriate, and put everything on one line even though the following text is word-wrapping):

ccollabupdatechangelist change-submit //depot/... "/usr/bin/ccollab --url <collabUrl> --user <collabUser> --password <collabPasswd> --scm perforce --p4port <p4port> --p4user <p4user> --p4passwd <p4passwd> --p4client <p4client> admin trigger update-changelist --description-prefix 'Review: ' --description-template 'ID ${review.id} by ${review.participants.rolename}' %changelist%

Optionally you can also add on the next line:

ccollabupdatechangelist form-in change "/usr/bin/ccollab --url <collabUrl> --user <collabUser> --password <collabPasswd> --scm perforce --p4port <p4port> --p4user <p4user> --p4passwd <p4passwd> admin trigger update-changelist --description-prefix 'Review: ' --description-template 'ID ${review.id} by ${review.participants.rolename}' %formfile%

© 2003-2011 SmartBear Software

Note - This example works under Windows, but under Unix the Perforce trigger system parses this line incorrectly, splitting up arguments by whitespace even though they are quoted using single-quotes.

The solution is to use a separate shell script file to call our code and to call that script from the Perforce trigger.

The --description-prefix prefix option is the prefix text that means: This is a line that should be updated with review information. The trigger is allowed to replace everything after that prefix with review information. If a changelist doesn't have a line with this prefix, a new line is appended to the end of the changelist description; otherwise the information is updated in-place.

The --description-template option is a template that will be inserted after the prefix. The template gives you a chance to decide which information about the review you want to include, in what order, and with any type of textual separators.

The template uses the ${...} variable substitution subsystem to let you insert dynamic data. In the example given above, the review ID and a list of participants is inserted between some English text.

If there is a review in the system that matches this changelist, the text is always updated either in-place or appended as described. On the other hand, if there is not already a review, the description is updated only if the prefix is already present; if there's no current review and no prefix than nothing is appended to the description.

4.4.7.27.8.6 create-review

```
Usage: ccollab [global-options] admin trigger create-review [--add-on-match
<value> --review-id-regex <value>] <changelist>
```

`create-review` - Creates a new Review for a changelist if no Review already exists

Command Options

Option	Required?	Description
--add-on-match <value>	N	If a review ID regex matches, add this changelist to the review
--review-id-regex <value>	N	Regular Expression that identifies Review ID in commit comment

© 2003-2011 SmartBear Software

Option	Required?	Description
<changelist>	Y	SCM-specific ID of an atomic set of changes

The create-review trigger automatically creates a review in PeerReview Complete after the change is committed to the Subversion server. It's smart enough to not create reviews if they've already been created for this code.

For this hook to work smoothly you will need your Subversion usernames and PeerReview Complete logins to match. Here "match" means "differs at most in capitalization." This ensures that when a developer checks in code the review is created under his PeerReview Complete account.

If you don't do this the hook will still work, but reviews will be created under the system administrator's account. Someone will have to log into the system with that account and assign the review to someone else.

You may optionally specify a regular expression that identifies a PeerReview Complete review ID inside a Subversion commit message. If you supply the regular expression, the create-review hook will check for this text in commit messages and automatically not create a new review if a non-canceled review with that ID already exists. This allows some users to do pre-commit review and others post-commit, and either way you know all code has been reviewed or at least that a review of all code exists in the system.

For Subversion, to install this trigger you will need to create a post-commit hook. If you already have a post-commit hook, you can add our tool wherever it is appropriate; otherwise you will need to create an executable hook as described in the Subversion documentation (typically a batch file under Windows or a shell script under Linux/Mac).

Example Windows batch file:

"C:\Program Files\PeerReview Complete Client\ccollab.exe" --url <collabUrl> --user <collabUser> --password <collabPasswd> --scm subversion --svn-repo-url svn://url/to/repo --svn-exe "C:\Program Files\Subversion\bin\svn.exe" admin trigger create-review --review-id-regex "review:\s+(\d+)" %2 || exit 1

Example Linux/OSX shell script:

/collab/install/ccollab --url <collabUrl> --user <collabUser> --password <collabPasswd> --scm subversion --svn-repo-url svn://url/to/repo --svn-exe /usr/bin/svn admin trigger create-review --review-id-regex "review:\s+(\d+)" $2 || exit 1

Note our use of "exit 1" to ensure that the script terminates with a non-zero exit code if our trigger application fails.

© 2003-2011 SmartBear Software

4.4.7.27.9 group

```
Usage: ccollab [global-options] admin group sub-command [sub-command-
options]
```

`group` - Administer Groups

Sub-Commands

Sub-Command	Description
create 272	Create a new Group
edit 273	Edit a Group
delete 274	Delete a Group
sync 274	Sync to supplied XML
member 276	Administer Group Members

4.4.7.27.9.1 create

```
Usage: ccollab [global-options] admin group create [--allow-associate-with-
reviews <value> --description <value> --enabled <value> --title <value>]
<guid>
```

`create` - Create a new Group

Command Options

Option	Required?	Description
--allow-associate-with-reviews <value>	N	Allow users to associate this group with reviews (y/n/yes/no/true/false/on/off)
--description <value>	N	
--enabled <value>	N	Enablement status (y/n/yes/no/true/false/on/off)

© 2003-2011 SmartBear Software

Option	Required?	Description
--title <value>	N	
<guid>	Y	GUID (Globally Unique IDentifier) of the group

Manually create a new group. You must be an Administrator to execute this command.

Creating a new group with this command is equivalent to manually creating the group using the web UI. The new group will not be modified by subsequent 'ccollab admin group sync' commands.

Examples:

ccollab admin group create qa

ccollab admin group create qa --title "Quality Assurance"

4.4.7.27.9.2 edit

```
Usage: ccollab [global-options] admin group edit [--allow-associate-with-
reviews <value> --description <value> --enabled <value> --title <value>]
<guid>
```

`edit` - Edit a Group

Command Options

Option	Required?	Description
--allow-associate-with-reviews <value>	N	Allow users to associate this group with reviews (y/n/yes/no/true/false/on/off)
--description <value>	N	
--enabled <value>	N	Enablement status (y/n/yes/no/true/false/on/off)
--title <value>	N	
<guid>	Y	GUID (Globally Unique IDentifier) of the group

© 2003-2011 SmartBear Software

Edit a group. You must be an Administrator to execute this command.

Examples:

ccollab admin group edit qa --description "Project Foo Quality Assurance"

ccollab admin group edit qa --title "Quality Assurance"

4.4.7.27.9.3 delete

```
Usage: ccollab [global-options] admin group delete <group>
```

`delete` - Delete a Group

Command Options

Option	Required?	Description
<group>	Y	GUID (Globally Unique IDentifier) of the group

Delete a group. You must be an Administrator to execute this command. Groups that are associated with reviews can not be deleted - try disabling them instead.

4.4.7.27.9.4 sync

```
Usage: ccollab [global-options] admin group sync [--create-users --delete-
groups] <syncfile>
```

`sync` - Sync to supplied XML

Command Options

Option	Required?	Description
--create-users	N	Whether to create users in the syncdata that don't already exist; if not specified, nonexistent users will be ignored

© 2003-2011 SmartBear Software

Option	Required?	Description
--delete-groups	N	Whether to delete groups not in the syncdata; if not specified, groups are disabled instead
<syncfile>	Y	Filename of the input XML syncdata, or - for stdin

Synchronize the server's group configuration to match that of the supplied XML. However, no changes will be made to manually-created groups. See the user's manual for documentation on the XML format. You must be an Administrator to execute this command.

Examples:

```
ccollab admin group sync /tmp/input --delete-groups
cat /tmp/input | ccollab admin group sync - --create-users
```

Example XML input:

```
<groups>
    <group guid="groupa" title="Group A" description="The A Group concerned with initials">
        <member-user login="alice" />
        <member-user login="adama" />
    </group>
    <group guid="groupb" title="Group B" description="The B Group that includes those in
Group A">
        <member-user login="bob" />
        <member-user login="betty" />
        <member-group guid="groupa" />
    </group>
    <group guid="groupc" title="Group C" description="The C Group that is everyone."
        includes-all-users="true" allow-associate-with-reviews="no">
        <member-group guid="groupb" />
    </group>
```

© 2003-2011 SmartBear Software

</groups>

See full XML Schema.

4.4.7.27.9.5 member

```
Usage: ccollab [global-options] admin group member sub-command [sub-
command-options]
```

`member` - Administer Group Members

Sub-Commands

Sub-Command	Description
add 276	Add a member to a Group
remove 277	Remove a member from a Group

```
Usage: ccollab [global-options] admin group member add [--admin <value> --
all-users --group <value> --user <value>] <guid>
```

`add` - Add a member to a Group

Command Options

Option	Required?	Description
--admin <value>	N	Login of the group admin user to specify
--all-users	N	Specify All Users
--group <value>	N	Guid of the group to specify
--user <value>	N	Login of the member user to specify
<guid>	Y	GUID (Globally Unique IDentifier) of the group

© 2003-2011 SmartBear Software

Add a member to a group. You must be an Administrator to execute this command.

Examples:

ccollab admin group member add qa --user joe@example.com

ccollab admin group member add corporate --group qa

```
Usage: ccollab [global-options] admin group member remove [--admin <value>
--all-users --group <value> --user <value>] <guid>
```

`remove` - Remove a member from a Group

Command Options

Option	Required?	Description
--admin <value>	N	Login of the group admin user to specify
--all-users	N	Specify All Users
--group <value>	N	Guid of the group to specify
--user <value>	N	Login of the member user to specify
<guid>	Y	GUID (Globally Unique IDentifier) of the group

Remove a member from to a group. You must be an Administrator to execute this command.

Examples:

ccollab admin group member remove qa --user joe@example.com

ccollab admin group member remove corporate --group qa

4.4.7.27.10 changelist

```
Usage: ccollab [global-options] admin changelist sub-command [sub-command-
options]
```

© 2003-2011 SmartBear Software

`changelist` - Administer Changelists

Sub-Commands

Sub-Command	Description
update-id 278	Updates the changelist ID so PeerReview Complete reflects the renumbered Perforce changelist ID

4.4.7.27.10.1 update-id

```
Usage: ccollab [global-options] admin changelist update-id <old-id> <new-id>
```

`update-id` - Updates the changelist ID so PeerReview Complete reflects the renumbered Perforce changelist ID

Command Options

Option	Required?	Description
<old-id>	Y	
<new-id>	Y	

Updates the changelist ID using the provided configuration to match an existing Version Control configuration on the PeerReview Complete server to update all references to the old changelist ID. For example:

ccollab --scm p4 --p4port p4server:1666 admin changelist update-id 123 147

4.5 Tray Notifier

The Tray Notifier is a cross-platform tray notifier client interface to the PeerReview Complete server. It can be used to provide an easy way to see what reviews you are currently participating in, and prompt notification when a review requires your attention.

Note: the Tray Notifier is currently not available for Mac OS X

© 2003-2011 SmartBear Software

Starting the Tray Notifier

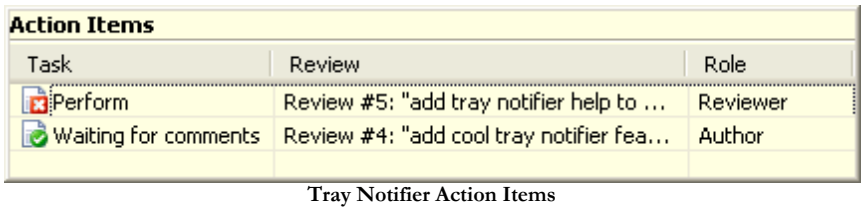
The Tray Notifier is an executable named `ccollabtray`. On Windows machines the graphical installer [183] includes an option to automatically run the Tray Notifier on startup (a shortcut to `ccollabtray` is placed in the Windows Start Menu Startup folder). Non-Windows platforms have to manually start `ccollabtray`.

Configuring the Tray Notifier

If you used the graphical installer [183], your connection to the PeerReview Complete Server should be configured already. Otherwise, you will be prompted when you try to connect. You can also select the *Preferences* item in the context menu [279] to open the preferences dialog. Use the *Server Connection Preferences* [187] page to configure the connection information to your PeerReview Complete Server.

Using the Tray Notifier

Clicking on the Tray Notifier opens a list of your current Action Items.



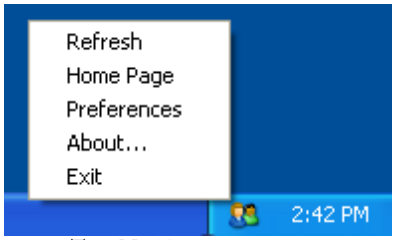
Tray Notifier Action Items

The Action Items list can be resized and moved by dragging it's borders. Action Items which are urgent appear with a red icon. Action Items which are not urgent appear with a green icon. Double-clicking on an Action Item opens a page in a browser window which lets you perform the action. Commit Action Items will open a wizard that will walk you through committing your files. Most changes can be committed with the tray notifier, but often changes to the directory structure will require that you use your SCM's tools directly.

The Tray Notifier icon changes to show a check-mark to alert you when there are new Action Items that require your attention.

Context Menu

The context menu of the Tray Notifier (right-click on Windows) includes several options:



Tray Notifier context menu

- *Refresh* immediately checks the PeerReview Complete Server for updated Action Items
- *Home Page* opens your PeerReview Complete Home Page [139] in a browser window

© 2003-2011 SmartBear Software

- *Preferences* opens the client preferences dialog 187
- *About...* tells you information about your installation
- *Exit* shuts down the Tray Notifier

4.6 External Diff Viewer launcher

The cross- platform External Diff Viewer launcher allows users to use a separate diff viewer of their choice to view diffs related to a particular review. The External Diff Viewer launcher ships with Source Gear DiffMerge 281 but also has preset arguments for the following:

- Araxis Merge
- IDM UltraCompare
- Guiffy
- Beyond Compare.

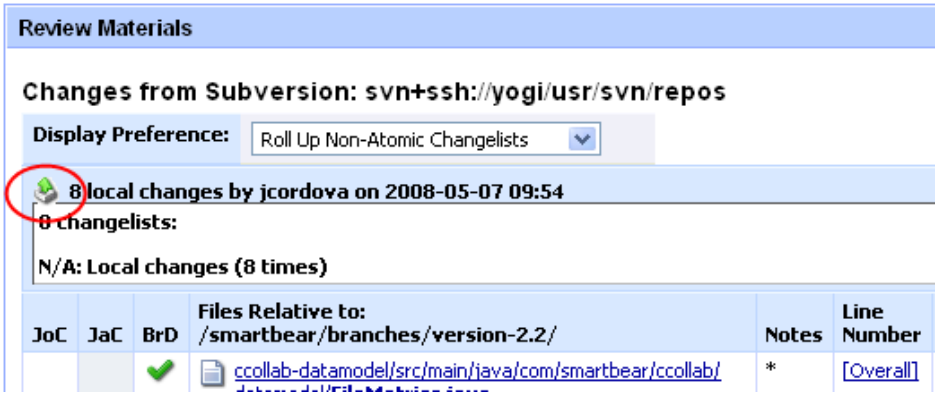
Launching External Diff for a single file

There is an External Diff button in the main diff viewer menu bar 159:

This button will launch the external diff viewer, which will compare the two versions of the single file shown in main diff viewer. You can change the versions shown to any option listed in the Compare 159 menu.

Launching External Diff for multiple files

There is an External Diff icon for each changelist listed in Review Materials 152:

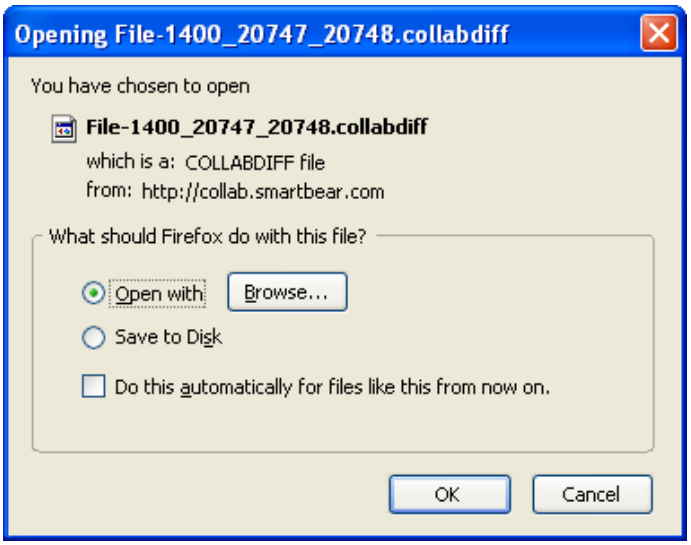


This will allow you to compare multiple files in the external diff viewer.

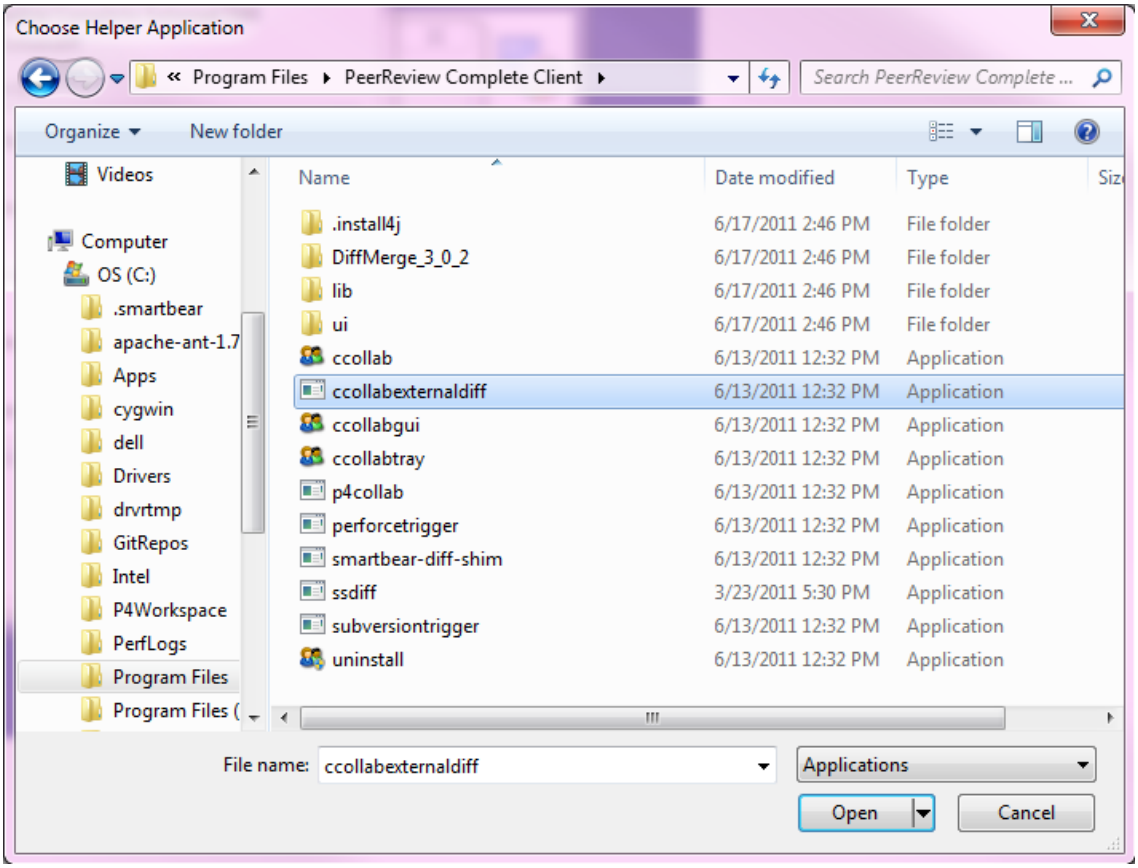
Starting the External Diff Viewer launcher

When first launching the External Diff Viewer launcher by clicking either of the two buttons above, you may be asked which application you'd like to open the file.

© 2003-2011 SmartBear Software



Choose "Open with" and click "Browse." You will be prompted to choose an application:

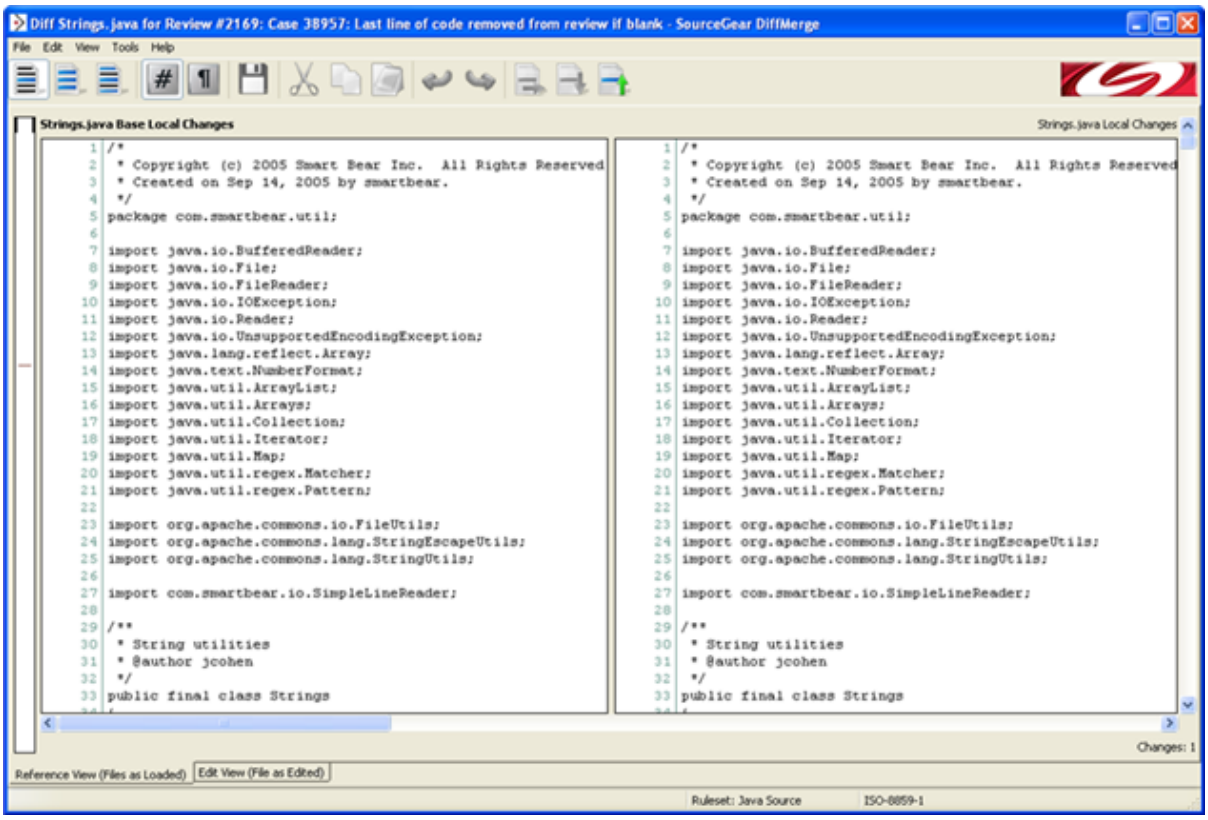


Choose the "ccollabexternaldiff.exe" under the "PeerReview Complete Client" folder, and click open. In the "Opening File" window, click "Do this automatically for files like this from now on" to save these settings, and click "OK."

The External Diff Viewer launcher will open Source Gear Diff Merge immediately. For more information on Diff Merge, visit http://sourcegear.com/diffmerge/.

© 2003-2011 SmartBear Software

Screen shot of single file comparisons in Diff Merge


Diff Strings.java for Review #2169: Case 38957: Last line of code removed from review if blank - SourceGear DiffMerge
File Edit View Tools Help
Strings.java Base Local Changes
Strings.java Local Changes
Changes: 1
Reference View (Files as Loaded)
Edit View (File as Edited)
Ruleset: Java Source
ISO-8859-1

Preferences

To configure your external diff viewer, visit thePreferences 194 menu in the GUI Client.

4.7 Print To Review

SmartBear Print To Review employs a Windows printer driver to submit documents for review. Windows applications can use Print To Review to add their document to a new or existing review. Documents are uploaded as images of the original document.

Features

- Any Windows application that can print is supported.
- Previously unsupported documents formats, e.g., Microsoft Office documents, can now be reviewed.

Compatibility

Windows XP SP2, Server 2003, Windows Vista and later Windows releases running on both 32-bit x86 and 64-bit hardware. Not all of these platforms are fully tested. Report any incompatibilities or problems to SmartBear. Please ask us if you're interested in support on other Windows platforms.

Print To Review requires the .NET 3.5 Framework. If it's not already installed on your system, the installer will prompt you to install the .NET framework first.

© 2003-2011 SmartBear Software

Image Quality

Like any printer driver, the image quality of documents uploaded for review is a function of the resolution and color depth. The default settings are for 150 DPI and 256 colors with halftoning. For improving font quality on text documents, 300 DPI black and white would be a better setting. For documents where the picture quality and color is important, 300 DPI and true color can be used. Keep in mind, however, that this latter setting can render extremely large documents. The driver compresses the image outputs in PNG format, but at that setting a busy color bitmap (i.e., without much compression) can easily generate image sizes upwards of 10MB per page.

Print To Review will not upload documents bigger than the client is allowed to upload (see upload truncation size in client properties 526). Excess pages will be replaced with a page indicating that the document has been truncated.

Installation

Make sure the GUI Client is installed 182 and configured 193. Then run the Print To Review standalone installer, available here.

If the Print To Review is installed to the same directory as the PeerReview Complete client, then the PATH environment variable will not need to be updated. Otherwise, the Print To Review installer will update the PATH to include the installed application folder, and will display a notification that a reboot is necessary before Print To Review can be used.

Uninstalling

Run the original installer, or uninstall via the Control Panel Add/Remove Programs menu. If a manual uninstall becomes necessary, refer to Advanced User Uninstall 285.

Important Notes for Vista and later Windows Platforms

Windows releases starting with Vista, including Windows 7 and Server 2008, exhibit some different behaviors than previous Windows releases. Most notably, Windows UAC (User Account Control) on these platforms introduces some security features which affect the install, uninstall, and operation of Print To Review.

Install and uninstall for Print To Review should be with an administrator account (not necessarily the Administrator user but a user with administrator privileges) or from a console window run as Administrator, for the most straightforward experience. Otherwise, depending on your UAC settings, you will be prompted during install and uninstall to run with Administrator privileges.

During uninstall, you may be prompted with a notification that the Print Spooler is using the files:

Select Continue at this prompt to uninstall the driver files.

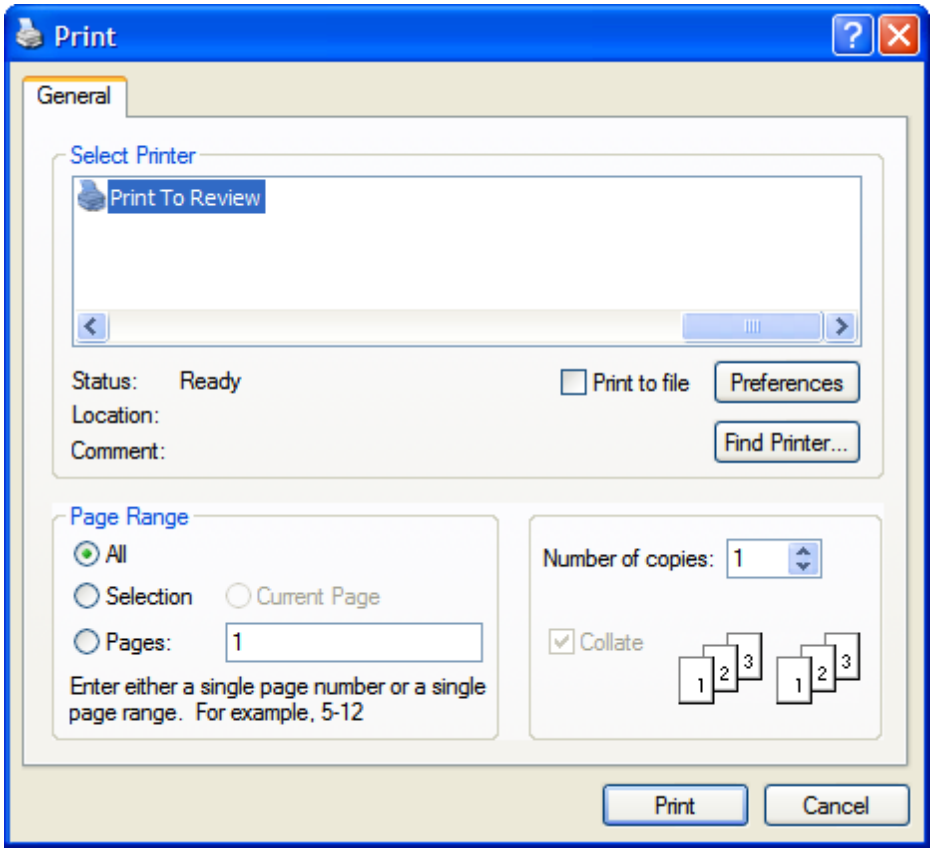
When uninstalling on Windows Vista or later, errors can occur if the spooler still has the driver files locked. This occurs when uninstalling within about a minute of printing to Print To Review and leaves Print To Review in an uninstallable state. The only simple recourse is to reinstall Print To Review and then uninstall again. Any of these steps will avoid this issue:

- Wait one minute after printing to Print To Review before uninstalling
- Stop then restart the spooler service before uninstalling
- Restart your system before uninstalling

© 2003-2011 SmartBear Software

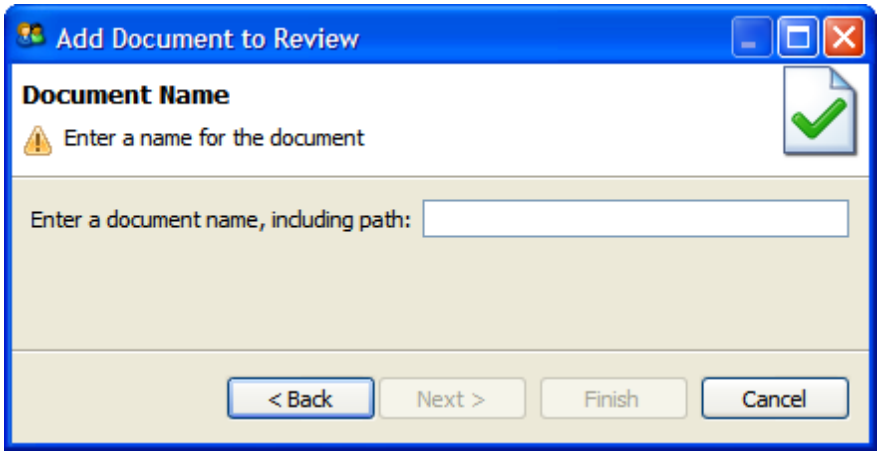
Upgrading
Print To Review does not support upgrades. To install a new version, please follow the uninstall instructions above, restart your system, and then install the desired version.

To Print To Review
From within your Windows application, click Print and select Print To Review as the Printer.



After printing your document, the GUI Client is launched to add the document to a review. On Windows Vista and later, depending on the UAC settings, GUI Client launches without foreground focus, and instead is displayed as a highlighted icon in the task bar and must be selected. Then choose a new or existing review, and next you will be prompted to enter a name for your document. This will be the name associated with the document images in the review:

© 2003-2011 SmartBear Software



Note that you may include path characters in the name, and the document name will be displayed with the path in the review. Once you've given the required information and click Finish, a browser window will open pointing to the review with your document images.

Limitations and Known Issues

Documents to be uploaded to the PeerReview Complete server are currently limited in size by the file truncation size defined in the Java VM Client Properties 526. Higher resolutions and color depths of Print To Review jobs increase the document image sizes uploaded for review. For documents that are too large, pages will be uploaded until the file truncation limit is encountered, and the remaining pages are discarded. Print To Review appends a page to these documents indicating the truncation. Printing at a lower resolution or in black and white is a workaround. Otherwise, increase the defined file truncation size.

Printing in a Remote Terminal environment currently cannot launch the GUI Client to print a document. These jobs are held in the print queue.

Troubleshooting

Windows XP SP2 and Server 2003

To submit problem reports or suggestions, open Add/Remove Programs from the Control Panel and select Support Information under Print To Review. Submit the displayed version number along with your description of the issue.

Windows Vista platforms and later

To submit problem reports or suggestion, open Programs and Features and find SmartBear Print To Review. The version number will be listed in the column labeled Version. If the Version column is not shown, click on the column headers and select to add the Version column. Submit the displayed version number along with your description of the issue.

Advanced User Uninstall

We of course recommend using Print To Review as installed by the installer. However, if it is customized in anyway (such as renaming the printer) or if a manual uninstall is otherwise desired, follow these steps.

Print To Review is installed as both an application and a printer driver. Manual uninstall of the driver can be done just as with any other printer driver. The print driver and port monitor can be deleted through the Server Properties dialog in the Print Spooler. The order of uninstalling the driver should be:

- Delete the printer

© 2003-2011 SmartBear Software

- Delete the printer driver
- Delete the CCOLLAB: port
- Uninstall the Print To Review application (this will generate errors but will complete)

4.8 Visual Studio Add-in

The PeerReview Complete Visual Studio Add-in allows files or changelists (for those SCM systems that support them) to be submitted for review from within the Visual Studio Solution Explorer, when the Solution, or when one or more Projects or Folders or Files are selected.

Features

- Integrates with the Solution Explorer in Visual Studio
- Creates or adds to reviews from locally modified files, or from changelists
- Integrates with the SCM system configured for the GUI Client 191

Compatibility
Visual Studio 2005, 2008 and 2010.

Installation
Make sure the GUI Client is installed 182 and configured for your PeerReview Complete server 193 and SCM system 191. Then run the Visual Studio Add-in standalone installer, available here.

The Visual Studio Add-in installer does not update the PATH environment variable to include the installed application folder (which is unnecessary if installing to the same application folder as the GUI Client). If installing to a different location, make sure to change your path to include the installed location.

The Visual Studio Add-in requires the .NET 3.5 Framework. If it's not already installed on your system, the installer will prompt you to install the .NET framework first.

To uninstall, run the installer again, or use the Control Panel Add/Remove Programs applet.

Usage
The Visual Studio Add-in is invoked from the Solution Explorer. A right mouse button click on a file, project or solution will bring up the context menu where you can select to Review changes 286 or Review changelists 286.

Review Changes
Selecting the `PeerReview Complete - Review changes` menu item will find modified files under the directory associated with the current Solution Explorer selection to be added to a new or existing review.

Review Changelists
Selecting the `PeerReview Complete - Review changelists` menu item will allow changelists associated with your configured SCM system to be added to a new or existing review.

© 2003-2011 SmartBear Software

Limitations and Known Issues
No validation is done for installing the`PeerReview Complete - Review changelists` menu item - not all SCM systems support changelists..

The installer does not update the PATH to include the installed application folder.

Troubleshooting
The Visual Studio Add-in shells out to the GUI Client 189 - if it is isn't already configured and working, the Visual Studio Add-in won't work. If you're experiencing issues with the Visual Studio Add-in, first verify the GUI Client works.

For us to look into an issue with Visual Studio Add-in, please enable debug logging and send us the resulting log file. Debug logging is enabled by editing the .config file for the Visual Studio Add-in and setting the Debug value to "1", i.e.:

```
<add key="Debug" value="1"/>
```

Both the .config file and the .log file reside in the "Smart Bear\Visual Studio Addin" subdirectory under the Common Application Data Folder, which, depending on your Windows platform, is typically one of these directories:

`"%ALLUSERSPROFILE%\Smart Bear\Visual Studio Addin"` on Windows 7 and Vista, or

`"%ALLUSERSPROFILE%\Application Data\Smart Bear\Visual Studio Addin"` on Windows XP

Note that the files in this directory do not have general user write permission under Vista and Windows 7. To modify the .config file, either run your editor with administrator privileges, or run the Visual Studio Add-in configuration utility with the "debug" command:

```
CcollabAddinConfig debug
```

Then restart Visual Studio, reproduce the issue, close Visual Studio, and find and send the debug log file and problem description to support@smartbear.com. The Visual Studio Add-in configuration utility also has a "nodebug" command to turn off debug logging.

We also welcome any suggestions about the current implementation - usability, appearance, functionality or other feedback that would help improve the Visual Studio Add-in.

To submit problem reports or suggestions, open Add/Remove Programs from the Control Panel and select Support Information under Visual Studio Add-in. Submit the displayed version number along with your description of the issue.

5 Version Control Integration

PeerReview Complete integrates with many popular Version Control Systems. Integrations include command-line, graphical, and Version Control triggers.

See the section below for your Version Control System:

- AccuRev 288
- Borland StarTeam 297

© 2003-2011 SmartBear Software

- CVS 302
- Git 311
- IBM CMVC 318
- IBM Rational ClearCase 322
- IBM Rational Synergy 336
- Mercurial 343
- Microsoft Team Foundation Server 349
- Microsoft Visual SourceSafe 356
- MKS Source Integrity 358
- Perforce 366
- SourceGear Vault 390
- Subversion 396
- Surround SCM 419

5.1 AccuRev Integration

This section describes PeerReview Complete integration with AccuRev:

GUI Client 289

The GUI Client can upload local changes to files 290 controlled by AccuRev before they are promoted. This includes both kept and modified files in the local Workspace. The GUI Client can also upload pending changes in a Stream 292, differences in a Transaction 290, and arbitrary AccuRev diffs 291.

Command Line Client 292

The Command Line Client can upload local changes to files 293 controlled by AccuRev before they are promoted. This includes both Kept and modified files in the local Workspace. The Command Line Client can also upload pending changes in a Stream 231, differences in a Transaction 294, and arbitrary AccuRev diffs 220.

Supported Versions

Our Command Line Client uses your own AccuRev command-line client (`accurev`) to communicate with the server. We support these versions:

- v4.5.3

Because we use client applications already present on your computer, we support all protocols, authentications, proxies, and other client configuration options you are currently using.

Support for Copy / Move

PeerReview Complete fully supports AccuRev's file and directory copy/move semantics. Thus, if a file is copied or moved rather than added from scratch, it will show up that way in the various user interfaces.

© 2003-2011 SmartBear Software

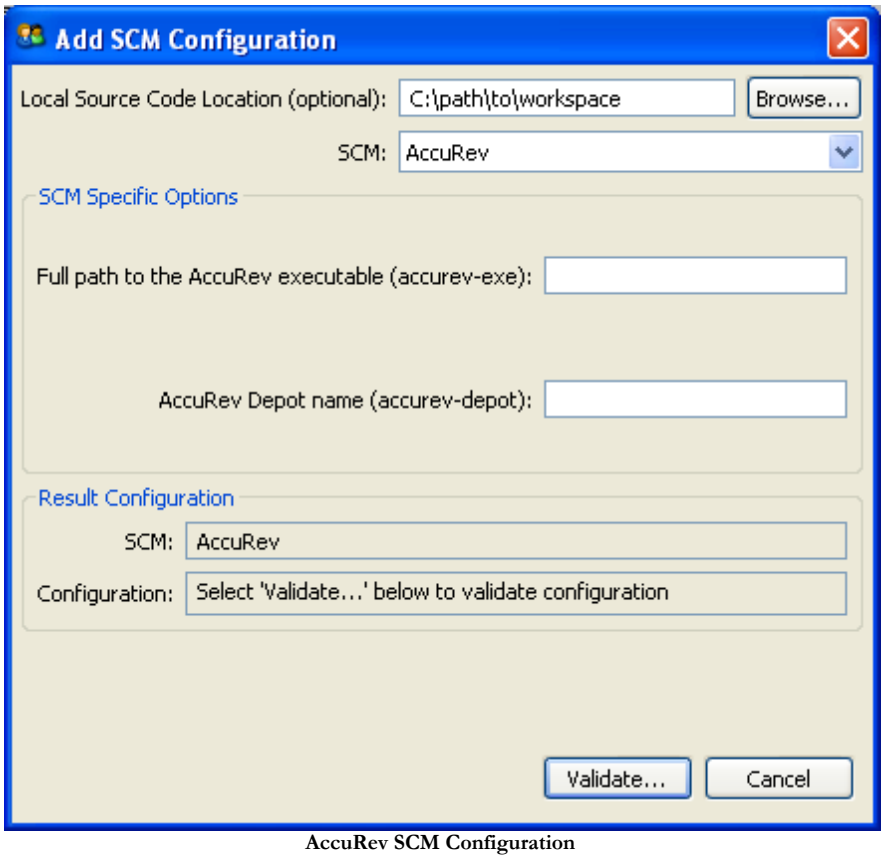
Support for Directory-level New/Delete/Copy/Move
PeerReview Complete partially supports AccuRev's concept of directories (not just files) being altered.

All files underneath the directories in question will be scanned, uploaded and represented properly in the GUI. The directories themselves will *not* be shown in any GUI, or even in the file list [200] confirmation screen presented by the command-line client.

This all works correctly even if, for example, you move a parent directory *and* alter/add/delete a file within that directory in the same changelist. You'll get the correct content for the file but the directory itself will not be listed in the review.

5.1.1 GUI Client

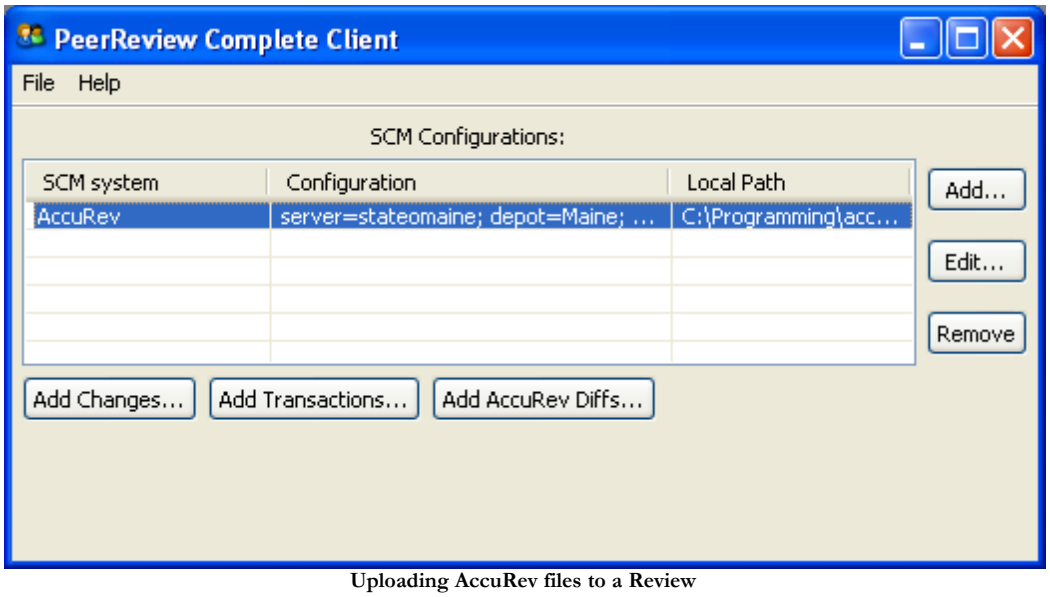
AccuRev-specific Options
The SCM Configuration dialog [191] has several AccuRev-specific options. If there is no AccuRev Workspace on your machine, you can still review committed AccuRev Transactions by specifying the AccuRev Depot name option.



AccuRev SCM Configuration

Uploading files to a Review
Selecting an AccuRev SCM Configuration in the GUI Client main screen [190] causes three Add to Review buttons to appear. The *Add Changes...* [290] button uploads modified files in a Workspace. The *Add Transactions...* [290] button uploads the files in committed Transactions. The *Add AccuRev Diffs...* [291] button uploads arbitrary diffs, or compares the differences between two Streams [292].

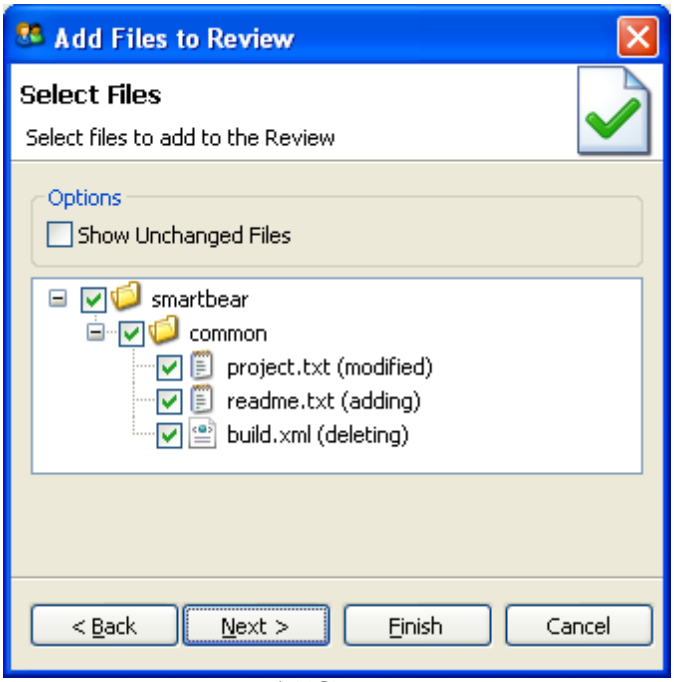
© 2003-2011 SmartBear Software



Uploading AccuRev files to a Review

Add Changes

Press the *Add Changes...* button to upload modified files in a Workspace to the PeerReview Complete Server for review.

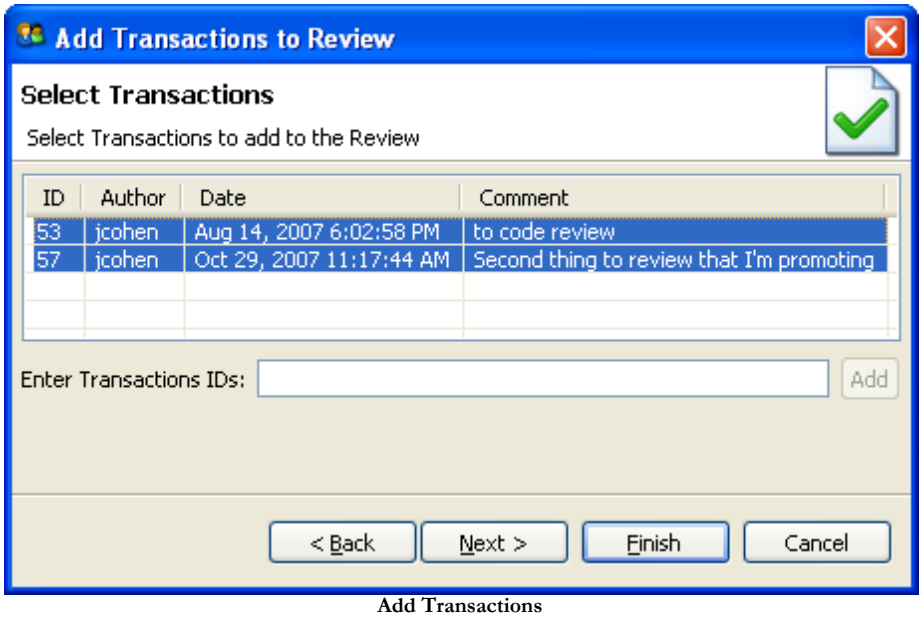


Add Changes

Add Transactions

Press the *Add Transactions...* button to upload the files in an AccuRev Transaction to the PeerReview Complete Server for review.

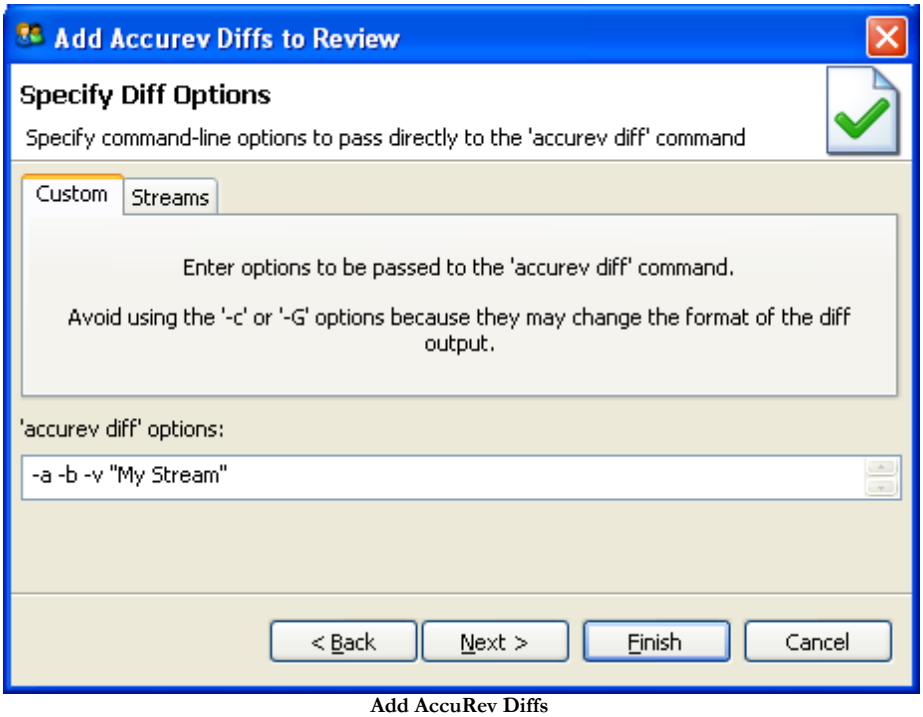
© 2003-2011 SmartBear Software



Add Transactions

Add AccuRev Diffs

Press the *Add AccuRev Diffs...* button to upload arbitrary AccuRev diffs to the PeerReview Complete Server for review.


Add Accurev Diffs to Review
Specify Diff Options
Specify command-line options to pass directly to the 'accurev diff' command
Custom
Streams
Enter options to be passed to the 'accurev diff' command.
Avoid using the '-c' or '-G' options because they may change the format of the diff output.
'accurev diff' options:
-a -b -v "My Stream"
< Back
Next >
Finish
Cancel

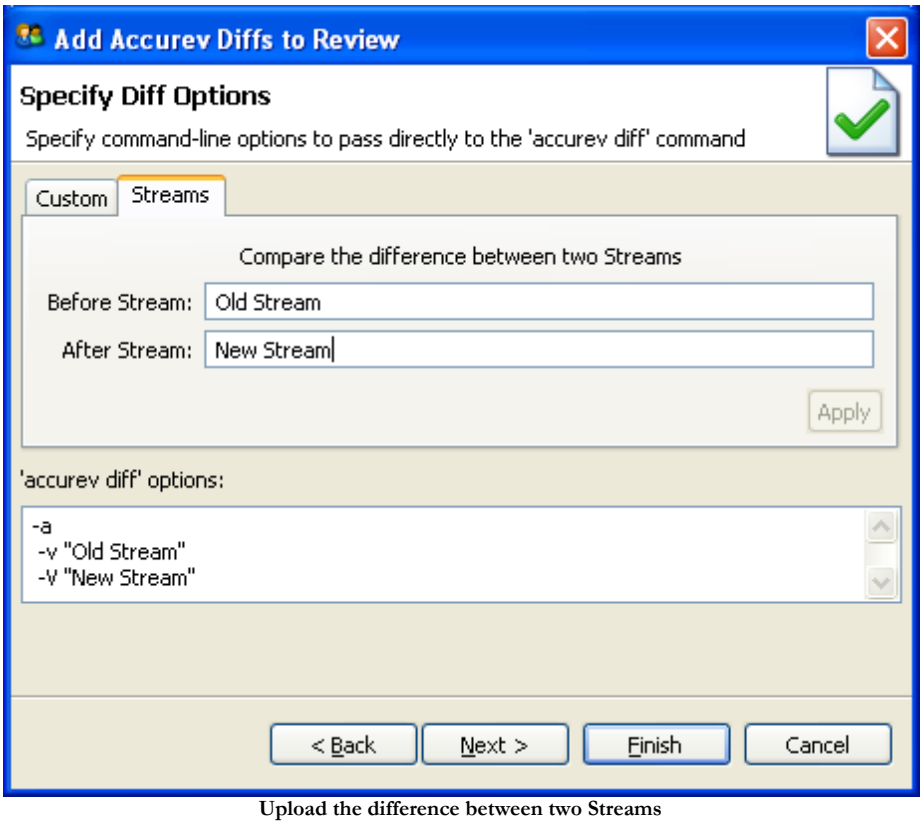
Add AccuRev Diffs

You can enter arbitrary AccuRev diff options, or select the Streams tab 292 to compare the difference between two Streams.

© 2003-2011 SmartBear Software

5.1.1.1 Comparing two Streams

Press the *Add AccuRev Diffs...* 291 button on the main screen 190 and then select the *Streams* tab to upload the difference between two Streams.



Upload the difference between two Streams

5.1.2 Command Line Client

Commands recommended for AccuRev

ccollab addchanges 293 - Attaches locally-modified files to a review

ccollab addchangelist 294 - Attaches an atomic changelist to a review

ccollab addardiffs 220 - Uploads diffs generated from accurev diff command

ccollab addstream 231 - Attaches pending differences from an AccuRev stream

The addchanges 293 command will upload local changes to files controlled by AccuRev before they are promoted. This includes both kept and modified files in the local workspace.

The addchangelist 294 command will upload a list of AccuRev transactions.

Configuration

In most cases, the Command Line Client can automatically detect your AccuRev configuration. Try testing your configuration 197 to verify the configuration is detected correctly.

© 2003-2011 SmartBear Software

If the Command Line Client is unable to detect your AccuRev configuration or you want to override the detected settings, you can manually specify AccuRev settings using global options 205.

To manually configure the Command Line Client to use AccuRev, execute the following command:

```
ccollab set 215 scm 205 accurev
```

AccuRev-specific Options

Option	Description
--accurev-depot <value>	AccuRev depot name
--accurev-exe <value>	Full path to the `accurev` command-line client
--accurev-anc-algorithm <value>	Which algorithm to use when calculating the predecessor, either predecessor, previous-occupant, or basis-version

Example:

```
ccollab set 215 accurev-depot MyDepot
```

5.1.2.1 addchanges (for AccuRev)

```
Usage: ccollab [global-options] addchanges [--diffbranch <value> --upload-
comment <value>] <review> <file-spec> [<file-spec> ...]
```

`addchanges` - Attaches locally-modified files to a review

Command Options

Option	Required?	Description
--diffbranch <value>	N	(ClearCase only) specify either branch-name or "predecessor" to determine the earlier version for upload
--upload-comment <value>	N	Comment used to upload files (default is "Local changes")
<review>	Y	Must be either an integer review-id, 'new', 'ask', or 'last'
<file-spec> [<file-spec> ...]	Y	Local files or directories to scan for modified files

© 2003-2011 SmartBear Software

Uploads locally modified files controlled by AccuRev to the PeerReview Complete server.

The first argument is the review specifier, then any number of files or directories can be given. Files are selected as-is; directories are scanned recursively for locally-modified files, which might include edits, additions, deletions, branches, integrations, moves, copies, and so on.

After the scan is complete, a file list is presented in a graphical editor so you can review the files about to be uploaded and make any corrections you wish.

For example, to create a new review and add all changes in the current directory and below, plus the file foo.txt, you would use:

```
ccollab addchanges new . foo.txt
```

To upload modified files from the current working directory and all subdirectories to review 123:

```
ccollab addchanges 123 .
```

To upload file foo.txt and modified files from c:\dev\project into a brand new review:

```
ccollab addchanges new foo.txt c:\dev\project
```

5.1.2.2 addchangelist (for AccuRev)

```
Usage: ccollab [global-options] addchangelist <review> <changelist>
[<changelist>  ...]
```

`addchangelist` - Attaches an atomic changelist to a review

Command Options

Option	Required?	Description
<review>	Y	Must be either an integer review-id, 'new', 'ask', or 'last'

© 2003-2011 SmartBear Software

Option	Required?	Description
<changelist> [<changelist> ...]	Y	SCM-specific ID of an atomic set of changes

Uploads Transactions from AccuRev to the PeerReview Complete server.

The first argument is the review specifier, subsequent arguments are the IDs of the Transactions to upload.

For example, to upload Transactions t4321 and t7568 to a new review:

```
ccollab addchangelist new 4321 7568
```

For example, to upload Transactions t5432 and t12654 to review 111:

```
ccollab addchangelist 111 5432 12654
```

5.1.2.3 addardiffs

```
Usage: ccollab [global-options] addardiffs [--upload-comment <value>]
<review> [<user-diff-arg> [<user-diff-arg> ...]]
```

`addardiffs` - Uploads diffs generated from accurev diff command

Command Options

Option	Required?	Description
--upload-comment <value>	N	Comment used to upload files (defaults to command-line arguments)
<review>	Y	Must be either an integer review-id, 'new', 'ask', or 'last'

© 2003-2011 SmartBear Software

Option	Required?	Description
<user-diff-arg> [<user-diff-arg> ...]	N	Command-line arguments to pass directly to the diff command

Upload differences between arbitrary versions of files in AccuRev using 'accurev diff'.

Do not use diff arguments that affect the diff output such as -c or -G. The PeerReview Complete command-line client will automatically select an output format that ensures you will get all the data you need on the server.

Examples:

ccollab addardiffs 698 -a -b -v "My Stream"

5.1.2.4 addstream

```
Usage: ccollab [global-options] addstream [--upload-comment <value>]
<review> [<stream>]
```

`addstream` - Attaches pending differences from an AccuRev stream

Command Options

Option	Required?	Description
--upload-comment <value>	N	Comment used to upload files
<review>	Y	Must be either an integer review-id, 'new', 'ask', or 'last'
<stream>	N	Specify an AccuRev stream by name, or leave blank to use the current workspace

Attaches differences pending promotion in an AccuRev stream.

© 2003-2011 SmartBear Software

5.2 Borland StarTeam Integration

This section describes PeerReview Complete integration with StarTeam:

GUI Client 399
The GUI Client can upload arbitrary StarTeam diffs 298, or the difference between two dates 298, Labels 299 or Revisions 300.

Command Line Client 301
The Command Line Client can upload arbitrary StarTeam diffs.

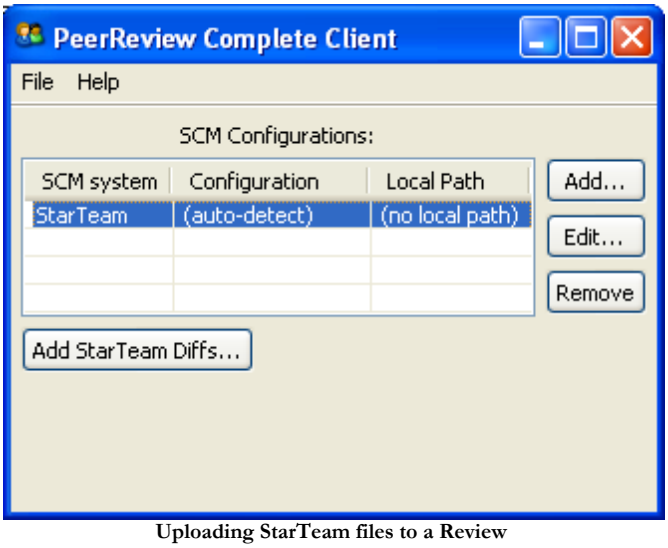
Supported Versions
Our integration uses your own installed StarTeam command line client executable (`stcmd.exe`) to generate differences for review. The StarTeam client provides full context lines of differences, which means the uploaded versions can contain the full file content of the previous and current file versions.

Because we use client applications already present on your computer, we support all protocols, authentications, proxies, and other client configuration options you are currently using.

5.2.1 GUI Client

StarTeam-specific Options
The SCM Configuration dialog 191 has no StarTeam-specific options.

Uploading files to a Review
Selecting a StarTeam SCM Configuration in the GUI Client main screen 190 causes the *Add StarTeam Diffs...* 298 button to appear. The *Add StarTeam Diffs...* 298 button uploads arbitrary diffs, or compares the difference between two dates 298, Labels 299 or Revisions 300.

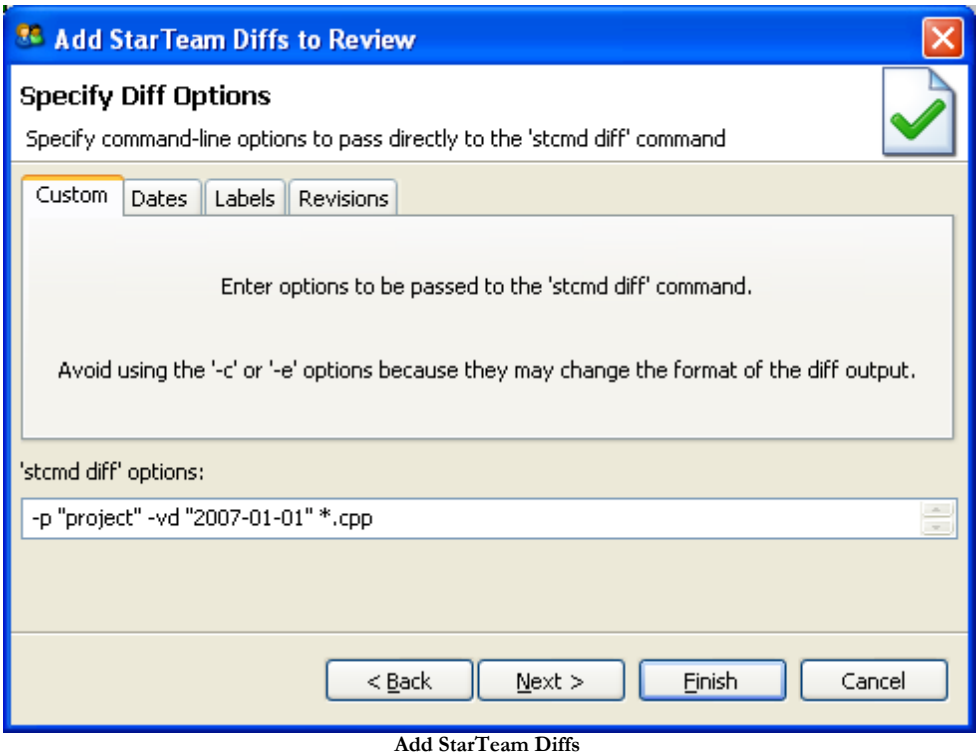


Uploading StarTeam files to a Review

© 2003-2011 SmartBear Software

Add StarTeam Diffs

Press the *Add StarTeam Diffs...* button to upload arbitrary StarTeam diffs to the PeerReview Complete Server for review.


Add StarTeam Diffs to Review
Specify Diff Options
Specify command-line options to pass directly to the 'stcmd diff' command
Custom
Dates
Labels
Revisions
Enter options to be passed to the 'stcmd diff' command.
Avoid using the '-c' or '-e' options because they may change the format of the diff output.
'stcmd diff' options:
-p "project" -vd "2007-01-01" *.cpp
< Back
Next >
Finish
Cancel

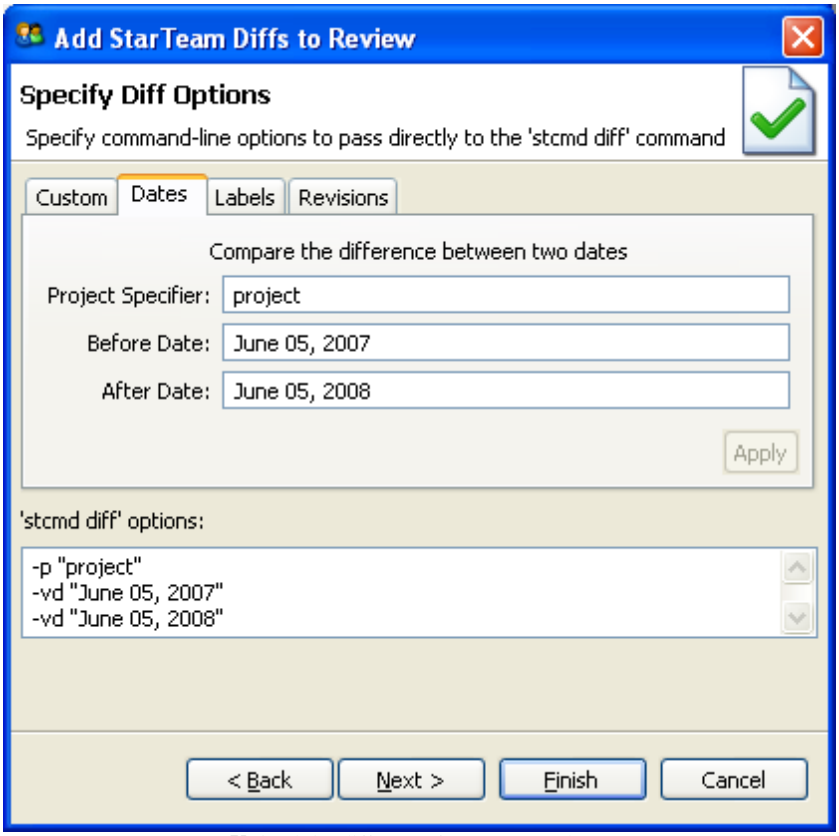
Add StarTeam Diffs

You can enter arbitrary StarTeam diff options, or compare the difference between two dates 298, Labels 299 or Revisions 300.

5.2.1.1 Comparing two dates

Press the *Add StarTeam Diffs...* 298 button on the main screen 190 and then select the *Dates* tab to upload the difference between two dates.

© 2003-2011 SmartBear Software

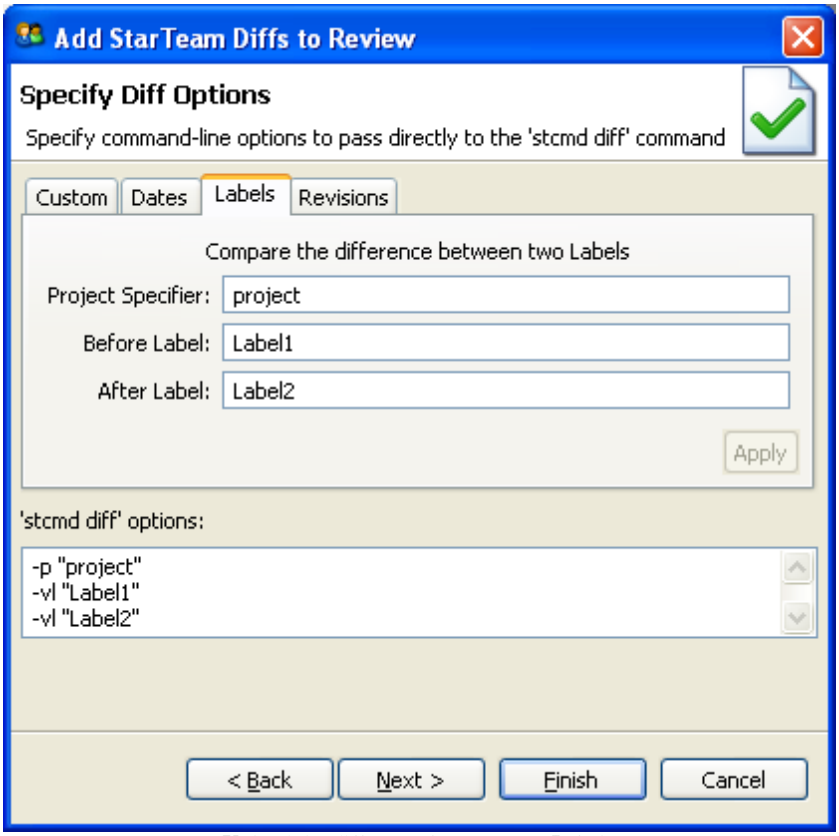


Upload the difference between two dates

5.2.1.2 Comparing two Labels

Press the *Add StarTeam Diffs...* 298 button on the main screen 190 and then select the *Labels* tab to upload the difference between two Labels.

© 2003-2011 SmartBear Software

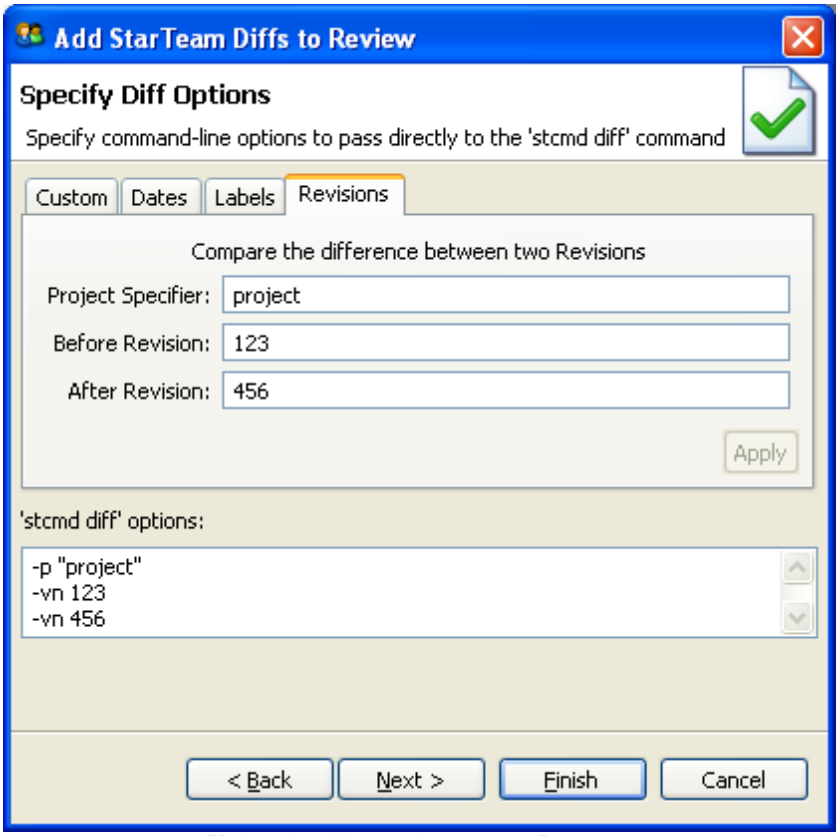


Upload the difference between two Labels

5.2.1.3 Comparing two Revisions

Press the *Add StarTeam Diffs...* 298 button on the main screen 190 and then select the *Revisions* tab to upload the difference between two Revisions.

© 2003-2011 SmartBear Software



Upload the difference between two Revisions

5.2.2 Command Line Client

Command recommended for StarTeam

ccollab addstdiffs 225 - Uploads diffs generated from the stcmd diff command

Configuration

No extra configuration is required to use StarTeam with the Command Line Client.

5.2.2.1 addstdiffs

```
Usage: ccollab [global-options] addstdiffs [--upload-comment <value>]
<review> [<user-diff-arg> [<user-diff-arg> ...]]
```

`addstdiffs` - Uploads diffs generated from the stcmd diff command

Command Options

© 2003-2011 SmartBear Software

Option	Required?	Description
--upload-comment <value>	N	Comment used to upload files (defaults to command-line arguments)
<review>	Y	Must be either an integer review-id, 'new', 'ask', or 'last'
<user-diff-arg> [<user-diff-arg> ...]	N	Command-line arguments to pass directly to the diff command

Upload differences between arbitrary versions of files in StarTeam using 'stcmd diff'.

Do not use diff arguments that affect the diff output such as -c or -e. The PeerReview Complete command-line client will automatically select an output format that ensures you will get all the data you need on the server.

Examples:

ccollab addstdiffs 698 -p "project" -vl LABEL1 -vl LABEL2

ccollab addstdiffs new -p "project" -vd "2007-01-01" *.cpp

5.3 CVS Integration

This section describes PeerReview Complete integration with CVS:

GUI Client 303
The GUI Client can upload local changes to files 304 controlled by CVS before they are checked into version control. The GUI Client can also upload arbitrary CVS diffs 304, or the difference between two Labels 305 or dates 306.

Command Line Client 307
The Command Line Client can upload local changes to files controlled by CVS before they are checked into version control. The Command Line Client can also upload arbitrary CVS diffs.

Supported Versions
Our Command Line Client 196 uses your own CVS command-line client (`CVS`) to communicate with the server. We support these versions:

- v1.11.1 and later
- v1.11.0 and before, only `ccollab addcvsdiffs` 221 is supported

© 2003-2011 SmartBear Software

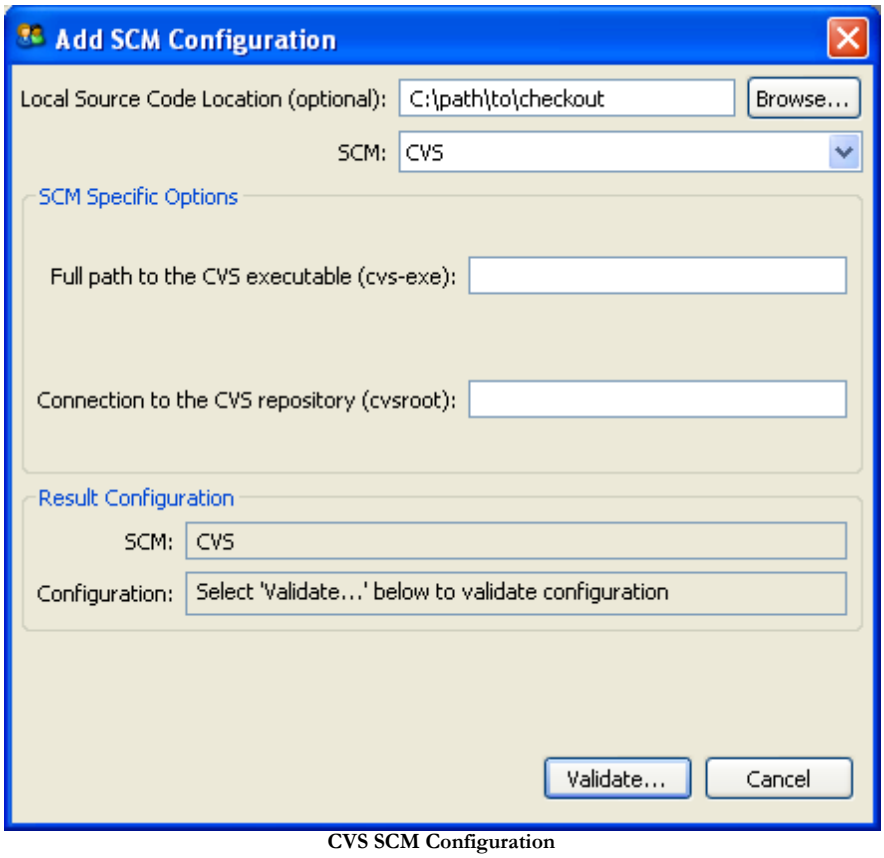
- `CvsNT` (Windows-only)

Because we use client applications already present on your computer, we support all protocols, authentications, proxies, and other client configuration options you are currently using.

5.3.1 GUI Client

CVS-specific Options

The SCM Configuration dialog 191 has several CVS-specific options.

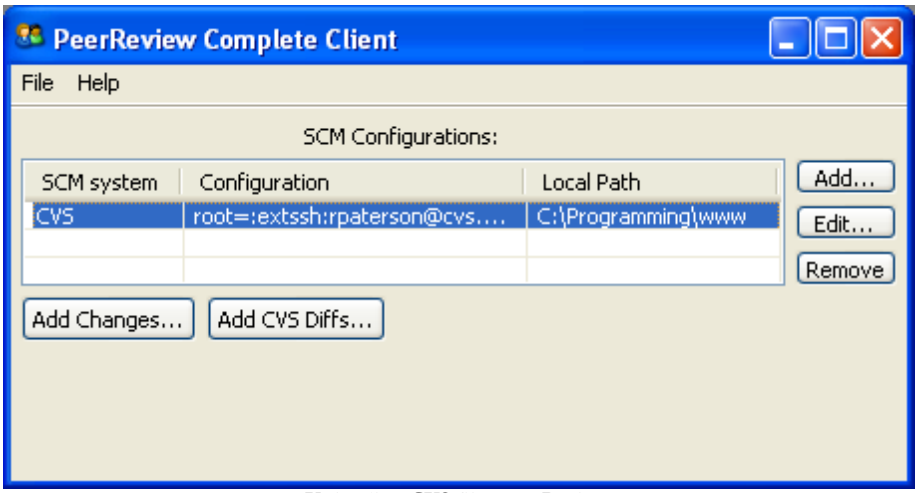


CVS SCM Configuration

Uploading files to a Review

Selecting a CVS SCM Configuration in the GUI Clientmain screen 190 causes several Add to Review buttons to appear. The *Add Changes...* 304 button uploads modified files in a CVS checkout. The *Add CVS Diffs...* 304 button uploads arbitrary diffs, or compares the difference between two Labels 305 or dates 306.

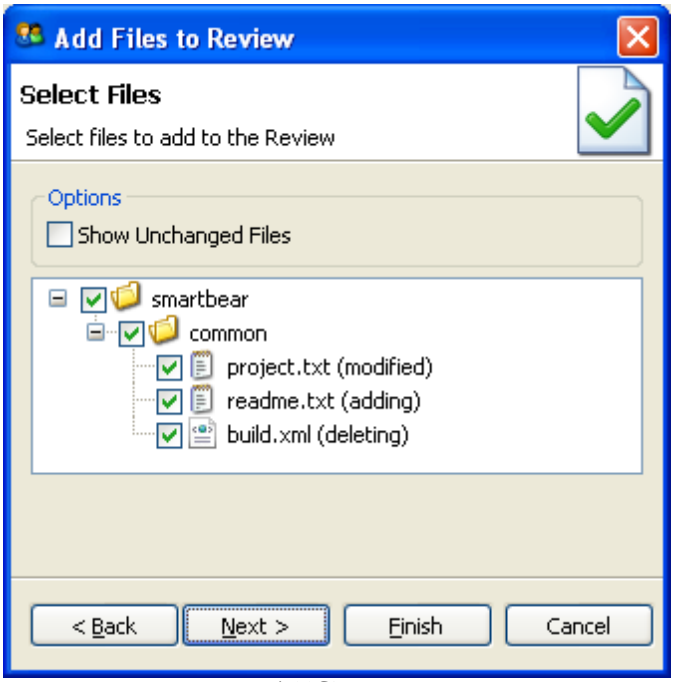
© 2003-2011 SmartBear Software



Uploading CVS files to a Review

Add Changes

Press the *Add Changes...* button to upload modified files in a CVS checkout to the PeerReview Complete Server for review.

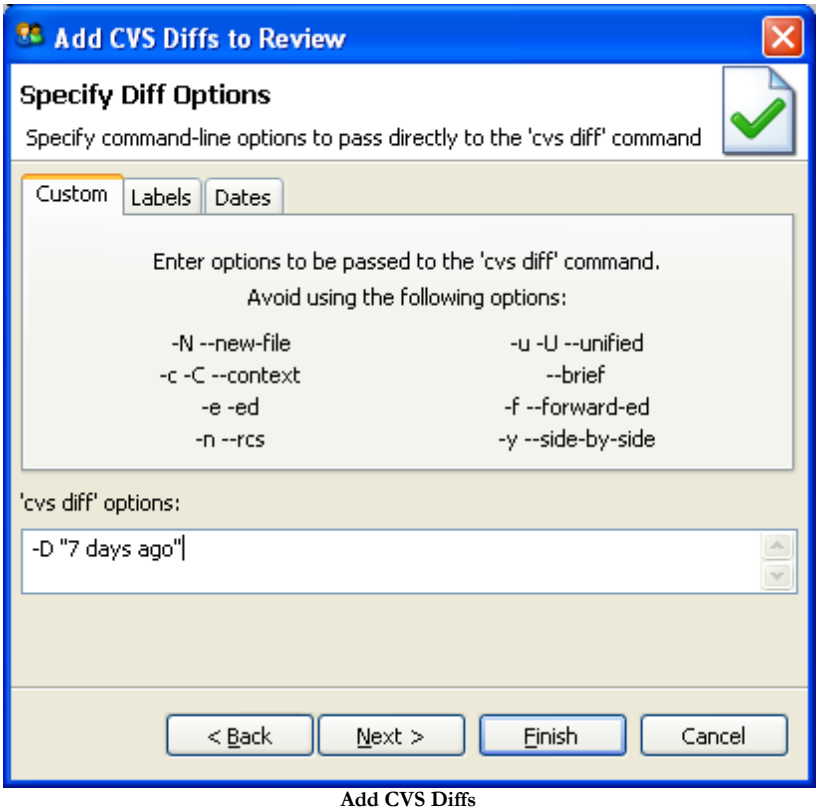


Add Changes

Add CVS Diffs

Press the *Add CVS Diffs...* button to upload arbitrary CVS diffs to the PeerReview Complete Server for review.

© 2003-2011 SmartBear Software



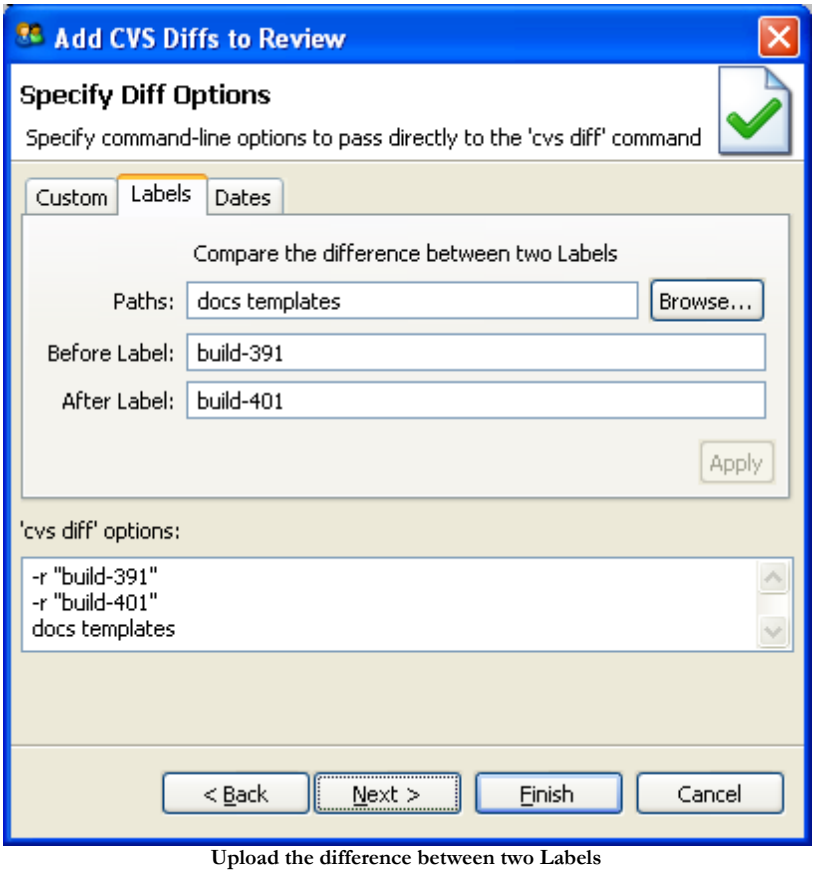
Add CVS Diffs

You can enter arbitrary CVS diff options, or compare the difference between two Labels 305 or dates 306 .

5.3.1.1 Comparing two Labels

Press the *Add CVS Diffs...* 304 button on the main screen 190 and then select the *Labels* tab to upload the difference between two Labels.

© 2003-2011 SmartBear Software

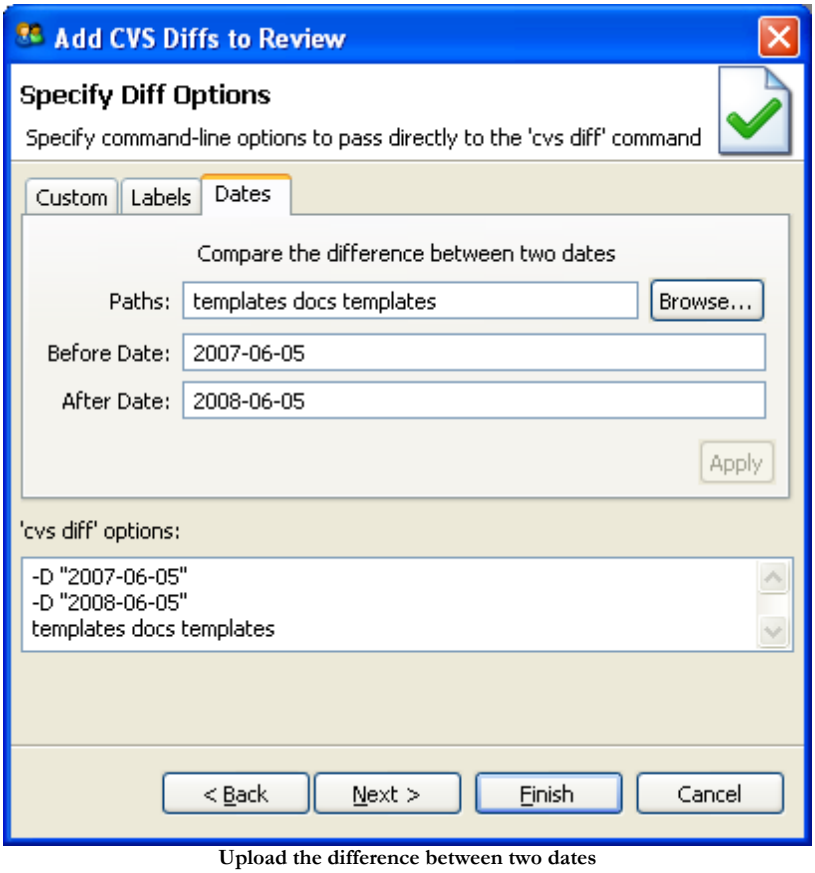


Upload the difference between two Labels

5.3.1.2 Comparing two dates

Press the *Add CVS Diffs...* 304 button on the main screen 190 and then select the *Dates* tab to upload the difference between two dates.

© 2003-2011 SmartBear Software



Upload the difference between two dates

5.3.2 Command Line Client

Commands recommended for CVS

ccollab addchanges 308 - Attaches locally-modified files to a review

ccollab addcvsdiffs 221 - Uploads diffs generated from cvs diff command

ccollab commit 310 - Commit changes in the review

The addchanges 308 command will upload local changes to files controlled by CVS before they are checked into version control.

Configuration

In most cases, the Command Line Client can automatically detect your CVS configuration. Try testing your configuration 197 to verify the configuration is detected correctly.

If the Command Line Client is unable to detect your CVS configuration or you want to override the detected settings, you can manually specify CVS settings using global options 206.

To manually configure the Command Line Client to use CVS, execute the following command:

```
ccollab set 215 scm 205 cvs
```

© 2003-2011 SmartBear Software

CVS-specific Options

Option	Description
--cvs-exe <value>	Full path to the `cvs` command-line client
--cvsroot <value>	Connection to the CVS repository

5.3.2.1 addchanges (for CVS)

```
Usage: ccollab [global-options] addchanges [--diffbranch <value> --upload-
comment <value>] <review> <file-spec> [<file-spec> ...]
```

`addchanges` - Attaches locally-modified files to a review

Command Options

Option	Required?	Description
--diffbranch <value>	N	(ClearCase only) specify either branch-name or "predecessor" to determine the earlier version for upload
--upload-comment <value>	N	Comment used to upload files (default is "Local changes")
<review>	Y	Must be either an integer review-id, 'new', 'ask', or 'last'
<file-spec> [<file-spec> ...]	Y	Local files or directories to scan for modified files

Uploads locally modified files controlled by CVS to the PeerReview Complete server.

The first argument is the review specifier, then any number of files or directories can be given. Files are selected as-is; directories are scanned recursively for locally-modified files, which might include edits, additions, deletions, branches, integrations, moves, copies, and so on.

After the scan is complete, a file list is presented in a graphical editor so you can review the files about to be uploaded and make any corrections you wish.

© 2003-2011 SmartBear Software

For example, to create a new review and add all changes in the current directory and below, plus the file foo.txt, you would use:

```
ccollab addchanges new . foo.txt
```

To upload modified files from the current working directory and all subdirectories to review 123:

```
ccollab addchanges 123 .
```

To upload file foo.txt and modified files from c:\dev\project into a brand new review:

```
ccollab addchanges new foo.txt c:\dev\project
```

5.3.2.2 addcvsdiffs

```
Usage: ccollab [global-options] addcvsdiffs [--upload-comment <value>]
<review> [<user-diff-arg> [<user-diff-arg> ...]]
```

`addcvsdiffs` - Uploads diffs generated from cvs diff command

Command Options

Option	Required?	Description
--upload-comment <value>	N	Comment used to upload files (defaults to command-line arguments)
<review>	Y	Must be either an integer review-id, 'new', 'ask', or 'last'
<user-diff-arg> [<user-diff-arg> ...]	N	Command-line arguments to pass directly to the diff command

Upload differences between arbitrary versions of files in CVS using 'cvs diff'.

© 2003-2011 SmartBear Software

Do not use diff arguments that affect the diff output such as -N --new-file, -u -U --unified, -c -C --context, --brief, -e -ed, -f --forward-ed, -n --rcs, or -y --side-by-side. The PeerReview Complete command-line client will automatically select an output format that ensures you will get all the data you need on the server.

For example, to upload all changes between labels build-391 and build-401:

```
ccollab addcvsdiffs review -r build-391 -r build-401
```

To upload all changes between dates 2006-01-01 and 2006-02-01:

```
ccollab addcvsdiffs review -D 2006-01-01 -D 2006-02-01
```

To upload all changes in the last 7 days:

```
ccollab addcvsdiffs review -D "7 days ago"
```

5.3.2.3 commit (for CVS)

```
Usage: ccollab [global-options] commit [--comment <value> --dismiss-only --
force] <review>
```

`commit` - Commit changes in the review

Command Options

Option	Required?	Description
--comment <value>	N	Comment for reviewed changes
--dismiss-only	N	Just dismiss the Action Item
--force	N	Ignore potential problems
<review>	Y	Must be either an integer review-id, 'ask', or 'last'

© 2003-2011 SmartBear Software

Commit the changes from a pre-commit review to sourcecontrol. Be sure to include a relevant comment.

5.4 Git Integration

This section describes PeerReview Complete integration with Git:

GUI Client 312
The GUI Client can upload arbitrary Git diffs and provides a special user interface for specifying branches as diff arguments. The GUI Client can also upload local changes to files that have been added to the Git index.

Command Line Client 315
The Command Line Client can upload arbitrary Git diffs and changes to files that have been added to the Git index.

Supported Versions
Our integration uses your own installed Git command line client executable to generate differences for review. We require Git v1.5.3 or later, as we use the --unified flag to provide full context lines of differences so that the uploaded versions can contain the full file content of the previous and current file versions.

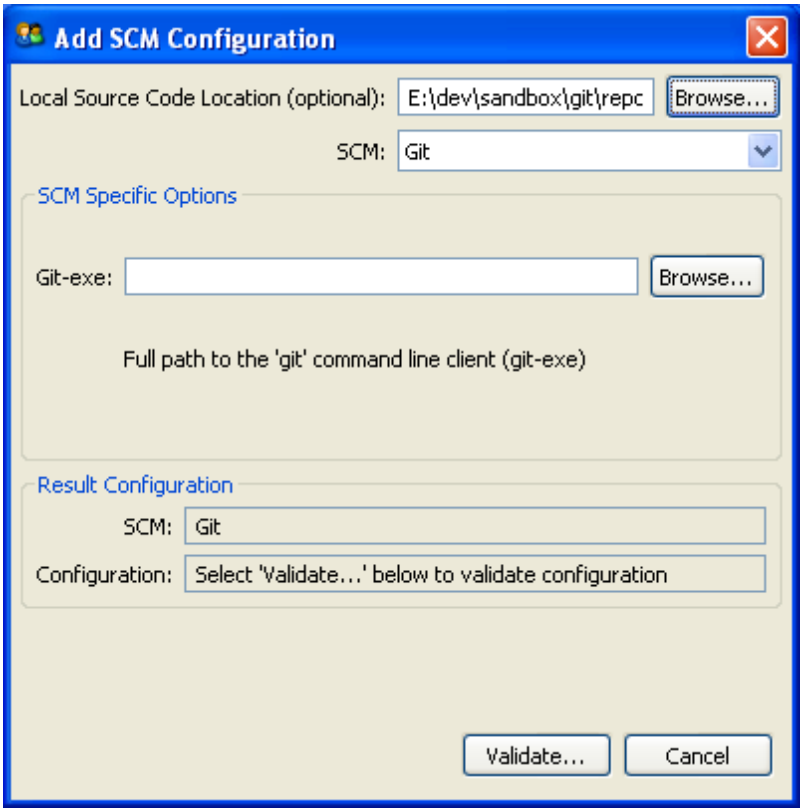
Because we use client applications already present on your computer, we support all protocols, authentications, proxies, and other client configuration options you are currently using.

© 2003-2011 SmartBear Software

5.4.1 GUI Client

Git-specific Options

The SCM Configuration dialog 191 has one Git-specific option. The full path to the Git command line executable on your local file system should be specified if it is not already on your system path. Note that for Git *Local Source Code Location* **is not** optional.

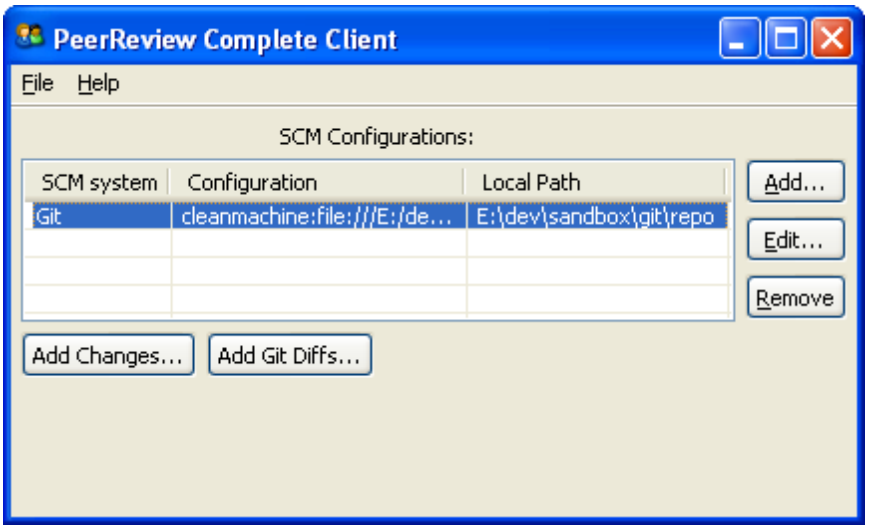


Configure client to work with Git

Uploading files to a Review

Selecting a Git SCM Configuration in the GUI Clientmain screen 190 causes *Add to Review buttons* to appear.

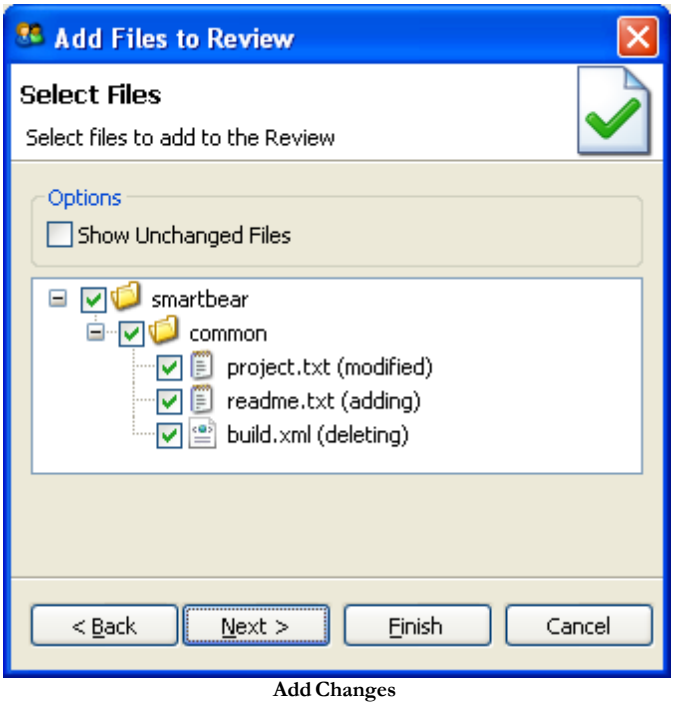
© 2003-2011 SmartBear Software



Uploading Git files to a review

Add Changes

Selecting *Add Changes...* allows you to upload the modifications that are currently in the index. These are the modifications that would be committed if you typed 'git commit' from a command line.

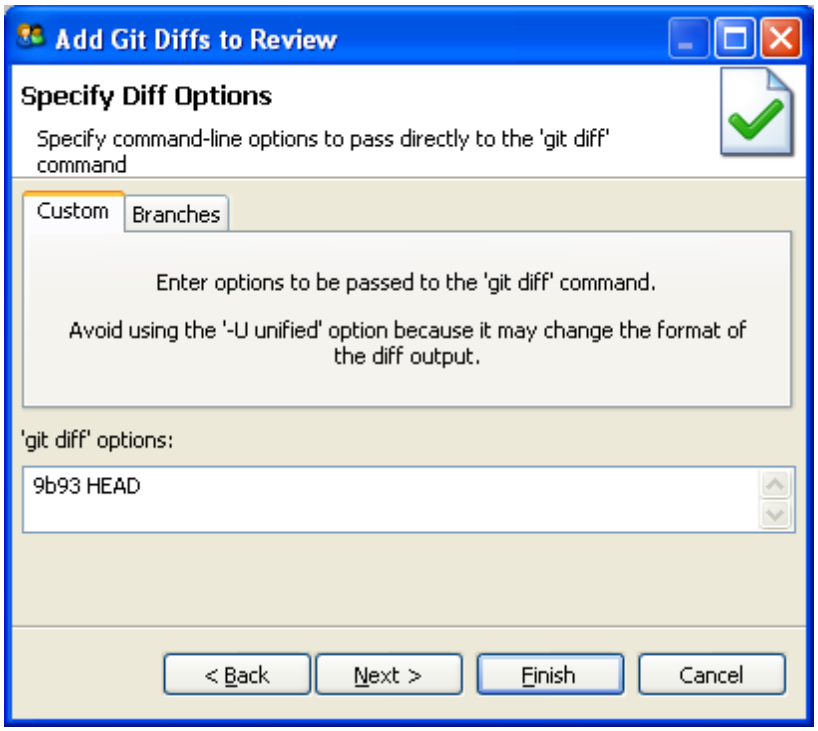


Add Changes

Add Git Diffs

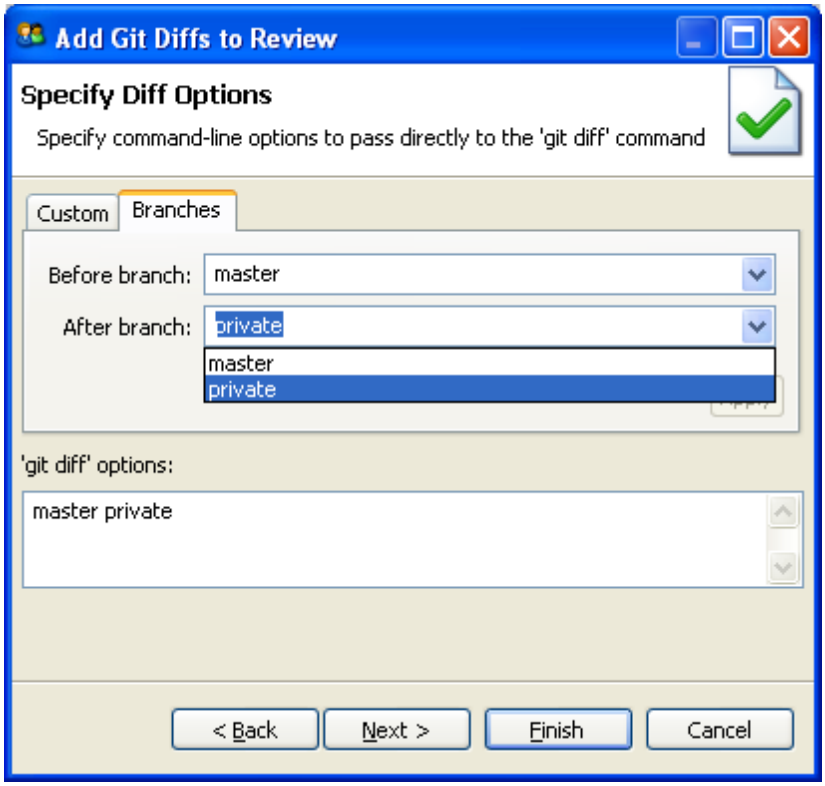
Press the *Add Git Diffs...* button to upload arbitrary Git diffs to the PeerReview Complete Server for review.

© 2003-2011 SmartBear Software


Add Git Diffs to Review
Specify Diff Options
Specify command-line options to pass directly to the 'git diff' command
Custom
Branches
Enter options to be passed to the 'git diff' command.
Avoid using the '-U unified' option because it may change the format of the diff output.
'git diff' options:
9b93 HEAD
< Back
Next >
Finish
Cancel

Upload arbitrary Git diffs

Press the *Add Git Diffs... button* and then click on the *Branches* tab to upload the differences between two branches to a review. Specify the branches by selecting from the *before* and *after* dropdown menus.

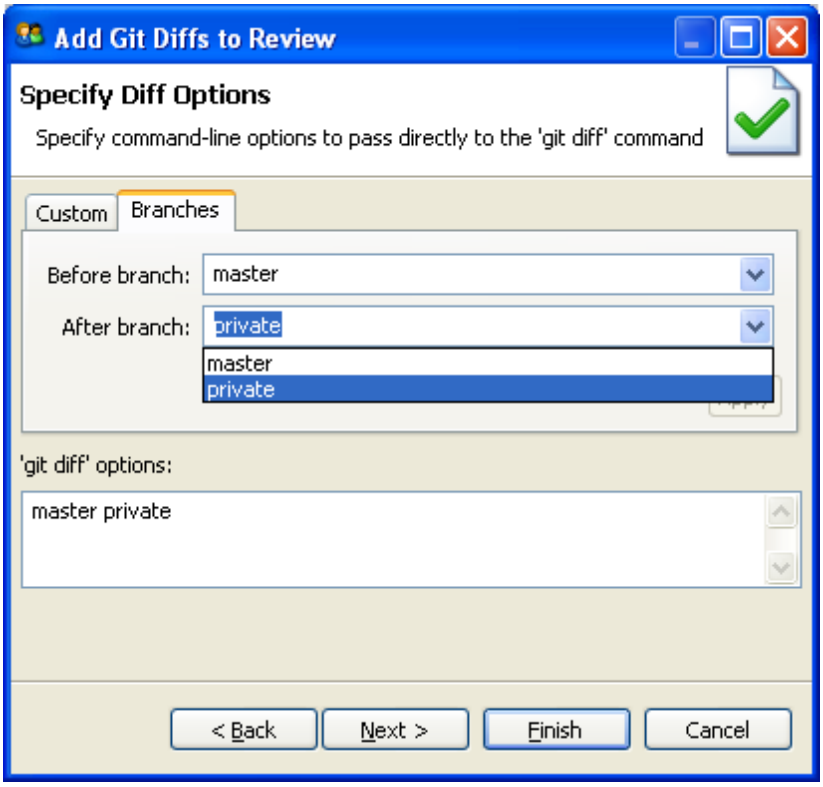


© 2003-2011 SmartBear Software

Upload the difference between two branches

5.4.1.1 Comparing two branches

Press the *Add Git Diffs...* 312 button on the main screen 190 and then select the *Branches* tab to upload the differences between two branches to a review. Specify the branches by selecting from the *before* and *after* dropdown menus.



Upload the difference between two branches

5.4.2 Command Line Client

Commands recommended for Git

ccollab addchanges 315 - Attaches locally-modified files to a review

ccollab addgitdiffs 222 - Uploads diffs generated by git diff command

Configuration

No extra configuration is required to use Git with the Command Line Client.

5.4.2.1 addchanges (for Git)

```
Usage: ccollab [global-options] addchanges [--diffbranch <value> --upload-
comment <value>] <review> <file-spec> [<file-spec> ...]
```

© 2003-2011 SmartBear Software

`addchanges` - Attaches locally-modified files to a review

Command Options

Option	Required?	Description
--diffbranch <value>	N	(ClearCase only) specify either branch-name or "predecessor" to determine the earlier version for upload
--upload-comment <value>	N	Comment used to upload files (default is "Local changes")
<review>	Y	Must be either an integer review-id, 'new', 'ask', or 'last'
<file-spec> [<file-spec> ...]	Y	Local files or directories to scan for modified files

Uploads locally modified files controlled by Git to the PeerReview Complete server.

The first argument is the review specifier, then any number of files or directories can be given. Files are selected as-is; directories are scanned recursively for locally-modified files, which might include edits, additions, deletions, branches, integrations, moves, copies, and so on.

After the scan is complete, a file list is presented in a graphical editor so you can review the files about to be uploaded and make any corrections you wish.

For example, to create a new review and add all changes in the current directory and below, plus the file foo.txt, you would use:

```
ccollab addchanges new . foo.txt
```

To upload modified files from the current working directory and all subdirectories to review 123:

```
ccollab addchanges 123 .
```

© 2003-2011 SmartBear Software

To upload file foo.txt and modified files from c:\dev\project into a brand new review:

```
ccollab addchanges new foo.txt c:\dev\project
```

5.4.2.2 addgitdiffs

```
Usage: ccollab [global-options] addgitdiffs [--upload-comment <value>]
<review> [<user-diff-arg> [<user-diff-arg> ...]]
```

`addgitdiffs` - Uploads diffs generated by git diff command

Command Options

Option	Required?	Description
--upload-comment <value>	N	Comment used to upload files (defaults to command-line arguments)
<review>	Y	Must be either an integer review-id, 'new', 'ask', or 'last'
<user-diff-arg> [<user-diff-arg> ...]	N	Command-line arguments to pass directly to the diff command

Upload differences between arbitrary versions of files in Git using 'git diff'.

Do not use diff arguments that affect the diff output such as -U unified. The PeerReview Complete command-line client will automatically select an output format that ensures you will get all the data you need on the server.

For example, to upload all changes between the revision 8 revisions ago and the revision 4 revisions ago:

```
ccollab addgitdiffs review master~8 master~4
```

To upload all changes in your local working directory:

© 2003-2011 SmartBear Software

ccollab addgitdiffs review

5.5 IBM CMVC Integration

This section describes PeerReview Complete integration with CMVC:

GUI Client 324
The GUI Client can scan CMVC for checked-out files and upload them before they are checked in.

Command Line Client 318
The Command Line Client can upload CMVC files both before or after they are checked in to CMVC.

Supported Versions
Our integration uses your own installed CMVC command line client executables (`defect.exe, track.exe, cmvclog.exe`) to communicate with the CMVC server. We support versions compatible with the CMVC95 command line spec.

Because we use client applications already present on your computer, we support all protocols, authentications, proxies, and other client configuration options you are currently using.

A Note about CMVC Tracks
Uploading tracks for review generates file differences based on Multiple Version Changelists 458.

5.5.1 GUI Client

TBD

5.5.2 Command Line Client

Commands recommended for CMVC

ccollab addchangelist 319 - Attaches an atomic changelist to a review

ccollab addversions 319 - Attaches any 2 given versions to a review

ccollab addtrack 230 - Attaches file parts for a Track associated with the given defect and release to a review

The addtrack 230 command can upload CMVC files both before or after they are checked in to CMVC.

Configuration
In most cases, the Command Line Client can automatically detect your CMVC configuration. Try testing your configuration 197 to verify the configuration is detected correctly.

If the Command Line Client is unable to detect your CMVC configuration you can manually specify CMVC using global options 204.

© 2003-2011 SmartBear Software

To manually configure the Command Line Client to use CMVC, execute the following command:

`ccollab set` 215 `scm` 205 `cmvc`

You must also be logged into CMVC in order to upload files to PeerReview Complete.

5.5.2.1 addchangelist (for CMVC)

```
Usage: ccollab [global-options] addchangelist <review> <changelist>
[<changelist> ...]
```

`addchangelist` - Attaches an atomic changelist to a review

Command Options

Option	Required?	Description
<review>	Y	Must be either an integer review-id, 'new', 'ask', or 'last'
<changelist> [<changelist> ...]	Y	SCM-specific ID of an atomic set of changes

Uploads Tracks from CMVC to the PeerReview Complete server.

The first argument is the review specifier, subsequent arguments are the IDs of the Tracks to upload.

For example, to upload Tracks 4321 and 7568 to a new review:

ccollab addchangelist new 4321 7568

For example, to upload Tracks 5432 and 12654 to review 111:

ccollab addchangelist 111 5432 12654

5.5.2.2 addversions (for CMVC)

```
Usage: ccollab [global-options] addversions [--upload-comment <value>]
<review> [<version-spec> [<version-spec> ...]]
```

© 2003-2011 SmartBear Software

`addversions` - Attaches any 2 given versions to a review

Command Options

Option	Required?	Description
--upload-comment <value>	N	Comment used to upload files (default is "Local changes")
<review>	Y	Must be either an integer review-id, 'new', 'ask', or 'last'
<version-spec> [<version-spec> ...]	N	Versions to be reviewed are given by <path> <version> [<previous version>], where <path> is the filename or server path of the file, <version> is the version to be reviewed, and <previous version> is an optional version that <version> should be diff'ed against. If <previous version> is not specified, the predecessor version of <version> will be used. If no <version-spec> is given on the command line, then one or more version specs will be read from stdin, one <version-spec> per line as described above. Note that arguments containing spaces must be quoted to be parsed correctly

Uploads specific versions of a file or files to a review. A version-spec is given by a filename followed by a version number, and optionally, a previous version number:

file-name version-number [previous-version-number]

If no previous version number is explicity given, diffs will be generated with the predecessor version as determined from the SCM system.

© 2003-2011 SmartBear Software

Adding versions for a single file can be done with command line arguments given after the review number. To upload versions for multiple files, the file and versions may be submitted on lines read from standard input. If the path and versions are not given on the command line, the client will expect to read them from stdin. Each line read should be in the format shown above. Some example command lines for reading versions from stdin:

ccollab addversions 86753
ccollab addversions last < versionlist.txt
cat versionlist.txt | ccollab addversions new

When supplying the version list in the command line client editor or an input file, the keyword 'local' can be used to denote the version corresponding to the local version of the file. The 'local' keyword can only be used for the first version argument, not the previous version.

5.5.2.3 addtrack

```
Usage: ccollab [global-options] addtrack [--upload-comment <value>]
<review> <defect-name> <release-name> [<root-path>]
```

`addtrack` - Attaches file parts for a Track associated with the given defect and release to a review

Command Options

Option	Required?	Description
--upload-comment <value>	N	Comment used to upload files (default is "Local changes")
<review>	Y	Must be either an integer review-id, 'new', 'ask', or 'last'
<defect-name>	Y	Associated CMVC defect
<release-name>	Y	Associated CMVC release

© 2003-2011 SmartBear Software

Option	Required?	Description
<root-path>	N	Path to local source files (CMVC top directory)

Upload files that are Parts in a CMVC Track to a review. If the root path is not specified, file content is extracted from CMVC, comparing the latest version in the Track to its predecessor version. If the root path is specified, the root path is used to resolve the file parts of the Track to the local files on disk, and these are compared against the file versions in the Track.

For example, to upload files that are already checked into CMVC:

```
ccollab addtrack review defect release
```

If you have a local snapshot directory with local modifications and the CMVC files exist and are part of a Track:

```
ccollab addtrack review defect release rootpath
```

rootpath specifies the root of the local sandbox that the file path names in the Track are relative to.

5.6 IBM Rational ClearCase Integration

This section describes PeerReview Complete integration with ClearCase:

GUI Client 324
The GUI Client can scan ClearCase for checked-out files and upload them before they are checked in 324. It upload changes from your activities 325., or changes from an activity or activities that you name 325. It can also upload arbitrary versions of ClearCase files after they are checked in.

Command Line Client 328
The Command Line Client can scan for checked-out files and upload them before they are checked in, upload arbitrary versions of ClearCase files after they are checked in, or upload UCM change sets for review.

Supported Versions
Our integration uses your own ClearCase command-line client (`cleartool`) to communicate with the server. We support these versions:

- Rational ClearCase and Rational ClearCase LT, versions 4.1 through 7.x

© 2003-2011 SmartBear Software

- Rational ClearCase Remote Client, version 7.1.2

Because we use client applications already present on your computer, we support all protocols, authentications, proxies, and other client configuration options you are currently using.

Support for ClearCase UCM
PeerReview Complete supports ClearCase UCM activities. The change set associated with a ClearCase activity can be added to a review using the Command Line Client 196 addactivity 229 command. For each file in the activity, the latest version in the activity and the predecessor of the earliest version in the activity are uploaded to the review.

A Note about ClearCase Activities
Uploading activities for review generates file differences based on Multiple Version Changelists 458.

Support for ClearCase Remote Client (CCRC)
Beta support for ClearCase Remote Client is now available! We welcome your feedback on how it works with your installation. If you run into errors, please create the appropriate debug log for your client and submit to support@smartbear.com.

PeerReview Complete integration with CCRC is done through the Rational Change Management Server API. During the PeerReview Complete client installation, there is an edit field allowing you to select the location of the Rational CM API jar files on your system. The required files are:

- remote_core.jar
- stpcc.jar
- stpwcmmn.jar
- stpwvcm.jar

These files are installed by the ClearCase Remote Client under the Rational Shared Resource Directory, e.g.:

```
C:\Program Files\IBM\IMShared\plugins\com.ibm.rational.teamapi_7.1.2.v201105310712
```

Specify the location of these files on your system so the installer can make them available to the PeerReview Complete client. Once installed, create a ClearCase SCM configuration using the the same connection parameters to the ClearCase CM server as your ClearCase Remote Client. These are your username and password, and the ClearCase TeamWeb services URL. For example, if your ClearCase server host name is cc-server, the default server URL would be:

```
http://cc-server:12080/TeamWeb/services/Team
```

Be careful to specify your connection parameters correctly. These parameters are not authenticated up front, instead calling back when server activity must take place. If the authentication is wrong, a cascade of errors can occur which can exhaust the RPC connections to the CM server.

© 2003-2011 SmartBear Software

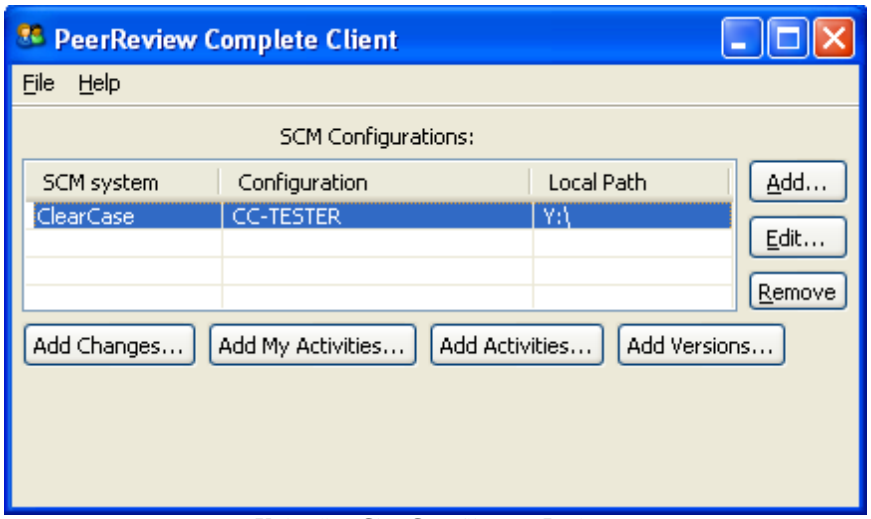
5.6.1 GUI Client

ClearCase-specific Options

The SCM Configuration dialog 191 has one ClearCase-specific option. Select the checkbox *Update snapshot view* if you want your ClearCase snapshot views to be updated prior to uploading files to a review.

Uploading files to a Review

Selecting a ClearCase SCM Configuration in the GUI Client main screen 190 enables several options for adding files for review. The *Add Changes...* 324 button finds modified files or allows selection of unmodified files, for uploading to a review. The *Add My Activities...* 325 button uploads changes from your activities. The *Add Activities...* 325 button uploads changes from an activity or activities that you name.

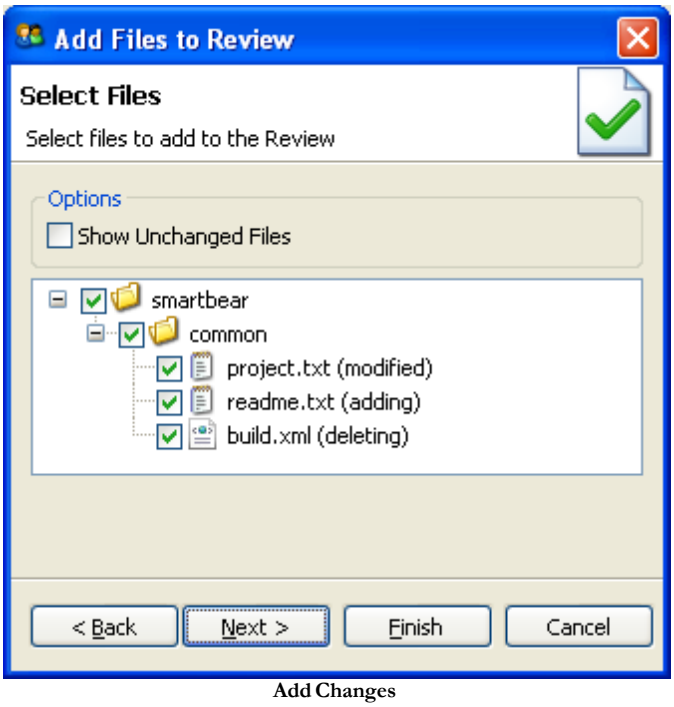


Uploading ClearCase files to a Review

Add Changes

Press the *Add Changes...* button to upload checked-out files to the PeerReview Complete Server for review.

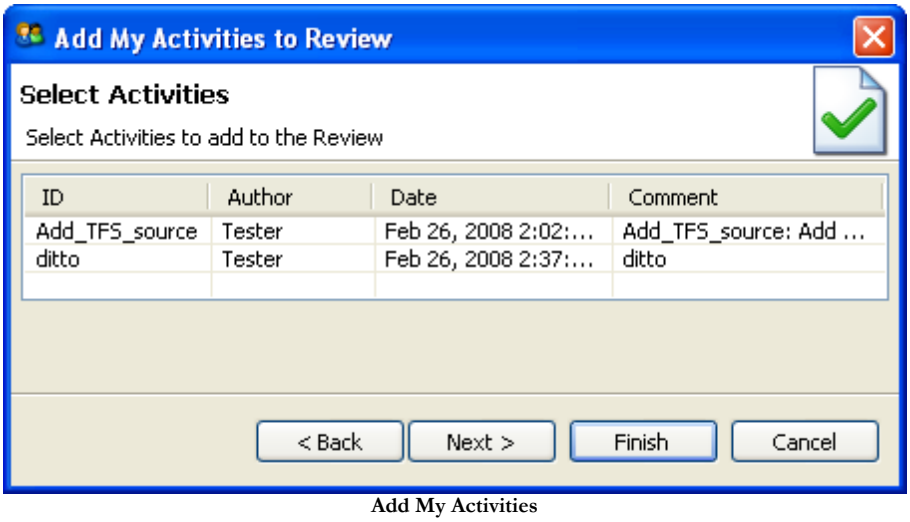
© 2003-2011 SmartBear Software



Add Changes

Add My Activities

Press the *Add My Activities...* button to upload files from selected activities which you own to the PeerReview Complete Server for review.

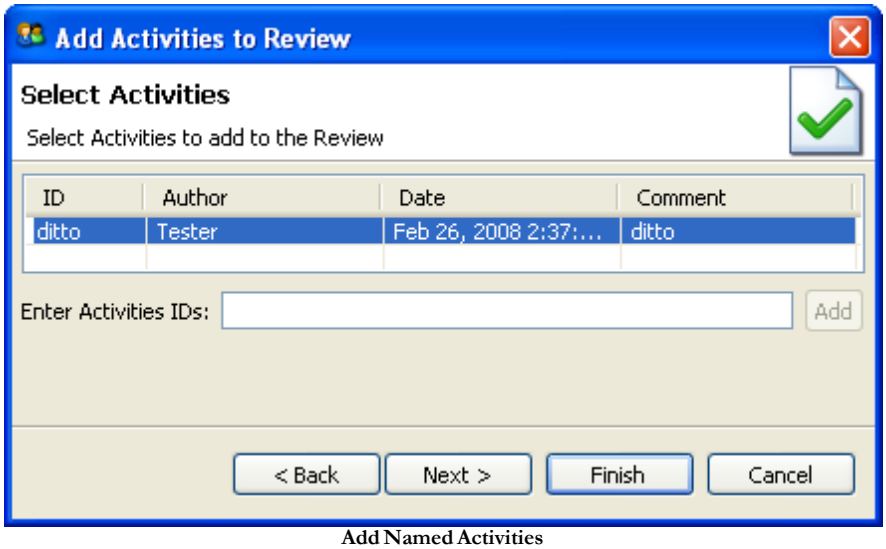


Add My Activities

Add Named Activities

Press the *Add Activities...* button to upload changes from a named ClearCase activity to the PeerReview Complete Server for review. You cannot upload activities that have CHECKEDOUT versions if you are not the activity owner.

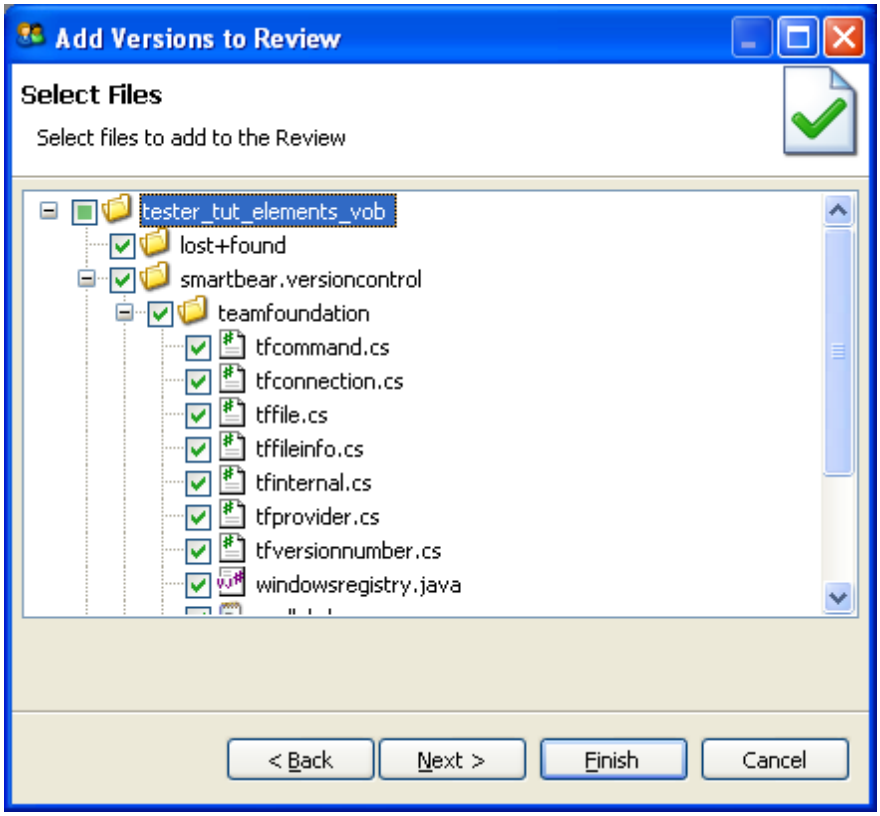
© 2003-2011 SmartBear Software



Add Named Activities

Add Versions

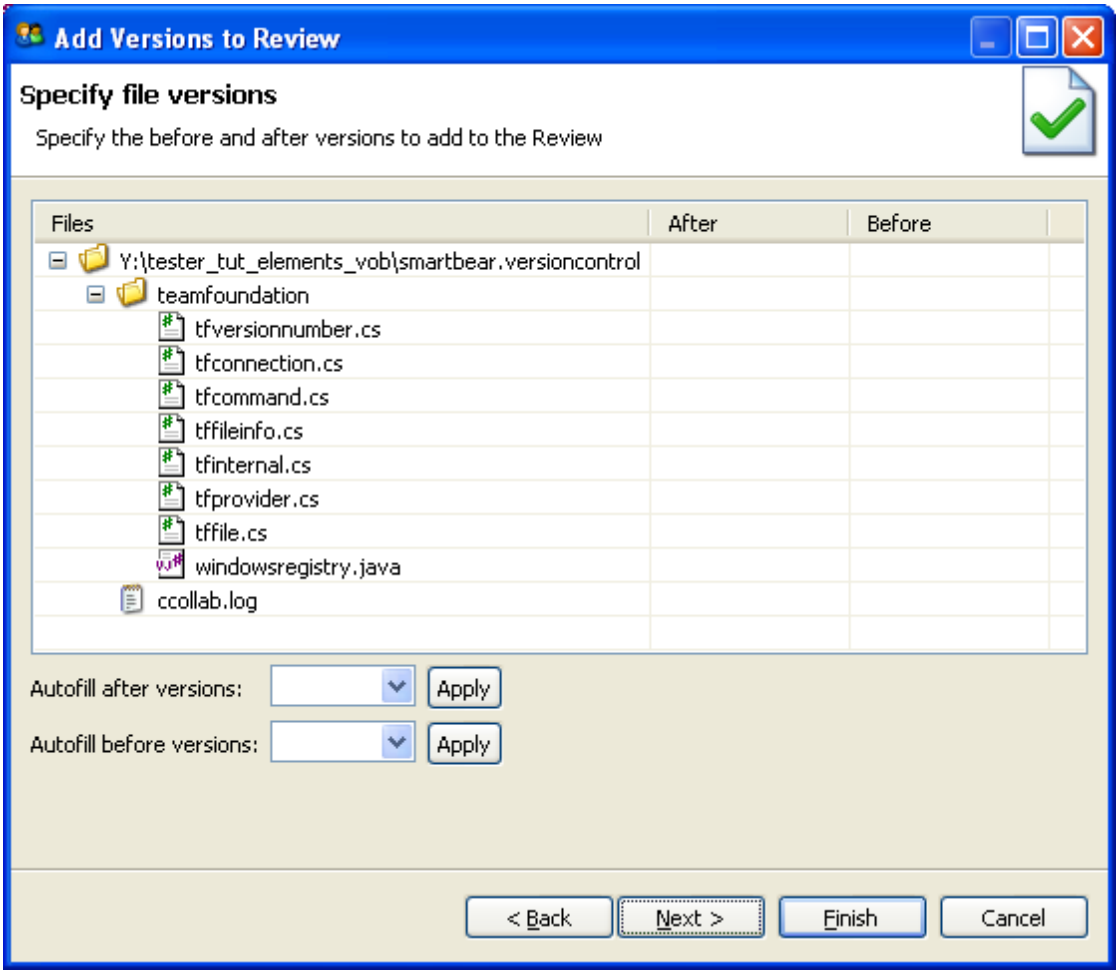
Press the *Add Versions...* button to upload versions of files in your working copy to the PeerReview Complete Server for review. Specify a name for the review or select an existing review and then click next. You will be prompted to select one or more file in your checkout (which was specified in theSCM Configuration dialog) 191. Do not press finish yet.



Select files

Click next so that you can specify versions of the files you just selected.

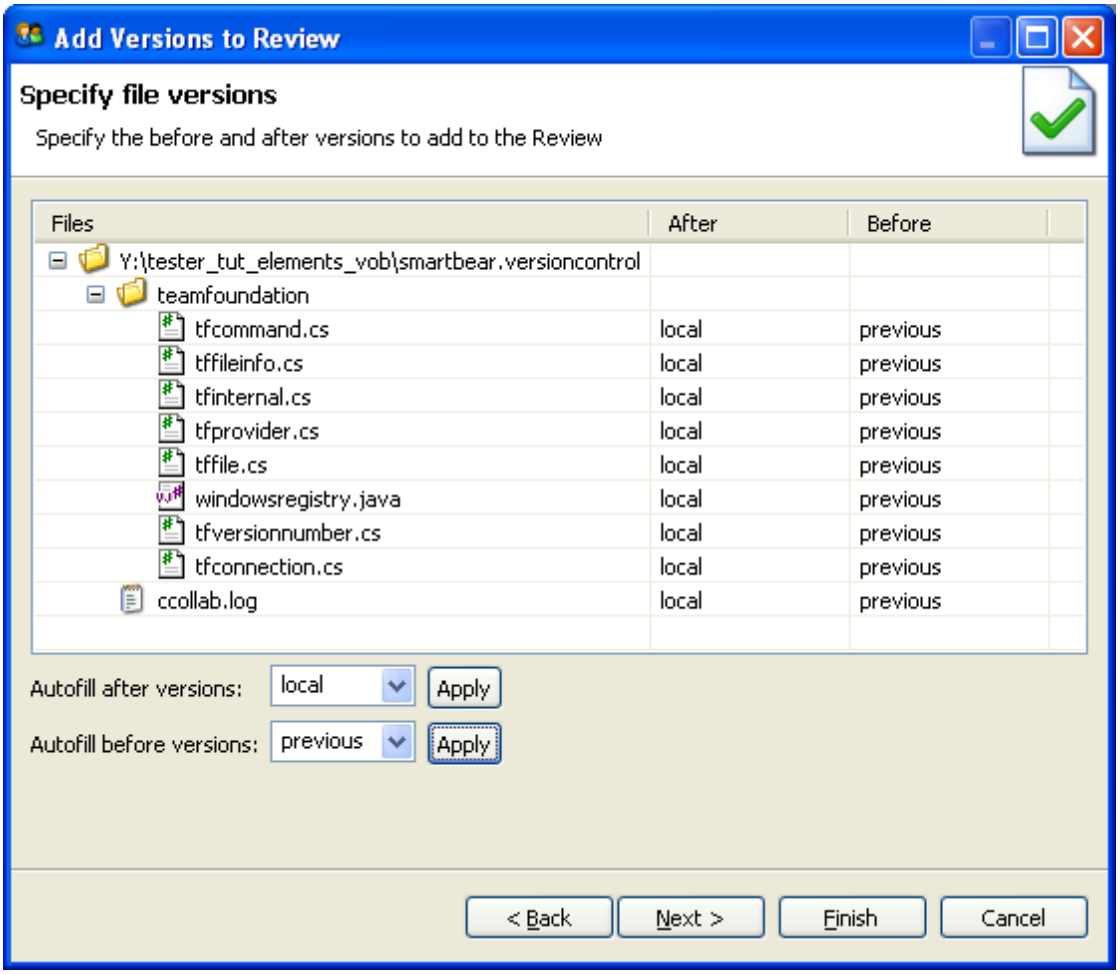
© 2003-2011 SmartBear Software



Specify versions

In the columns labeled before and after, enter the ClearCase versions of each file. Rows corresponding to directories are of course not editable. Certain keywords can be used in place of a ClearCase version. These are described more here: addversions 227. These keywords can be automatically filled into the before and after columns by selecting them from the dropdown menus and clicking apply. Each column can be cleared out by deleting the selected text in the dropdown and clicking apply.

© 2003-2011 SmartBear Software



Automatically fill in before/after values

5.6.2 Command Line Client

Commands recommended for ClearCase

ccollab addchanges 329 - Attaches locally-modified files to a review

ccollab addchangelist 331 - Attaches an atomic changelist to a review

ccollab addversions 332 - Attaches any 2 given versions to a review

ccollab addactivity 229 - Attaches file versions in a ClearCase activity to a review

ccollab commit 335 - Commit changes in the review

The addchanges 293 command will scan for checked-out files and upload them before they are checked in.

There is an option for the addchanges 329 command that allows local changes to be uploaded with a predecessor version in another branch as the previous version. The -diffbranch 329 option is used for this:

© 2003-2011 SmartBear Software

```
ccollab addchanges --diffbranch branch-name review file-or-dir
[file-or-dir] ...
```

For example, to create a new review and add all changes in the current directory and below, and compare them to the predecessor versions in the \main branch, you would use:

```
ccollab addchanges --diffbranch main new .
```

You can also use the special branch name `pre` to refer to the predecessor branch. For example, to add the local changes to hello.c compared to the most recent predecessor version in the previous branch to review 45, you would use:

```
ccollab addchanges --diffbranch pre 45 hello.c
```

The addversions [227] command will upload arbitrary versions of ClearCase files after they are checked in.

The addactivity [229] command will upload UCM change sets for review.

Configuration

In most cases, the Command Line Client can automatically detect your ClearCase configuration. Try testing your configuration [197] to verify the configuration is detected correctly.

If the Command Line Client is unable to detect your ClearCase configuration or you want to override the detected settings, you can manually specify ClearCase settings using global options [204].

To manually configure the Command Line Client to use ClearCase, execute the following command:

`ccollab set` [215] `scm` [205] `clearcase`

5.6.2.1 addchanges (for ClearCase)

```
Usage: ccollab [global-options] addchanges [--diffbranch <value> --upload-
comment <value>] <review> <file-spec> [<file-spec> ...]
```

`addchanges` - Attaches locally-modified files to a review

Command Options

Option	Required?	Description
--diffbranch <value>	N	(ClearCase only) specify either branch-name or "predecessor" to determine the earlier version for upload
--upload-comment <value>	N	Comment used to upload files (default is "Local changes")

© 2003-2011 SmartBear Software

Option	Required?	Description
<review>	Y	Must be either an integer review-id, 'new', 'ask', or 'last'
<file-spec> [<file-spec> ...]	Y	Local files or directories to scan for modified files

Uploads locally modified files controlled by ClearCase to the PeerReview Complete server.

The first argument is the review specifier, then any number of files or directories can be given. Files are selected as-is; directories are scanned recursively for locally-modified files, which might include edits, additions, deletions, branches, integrations, moves, copies, and so on.

After the scan is complete, a file list is presented in a graphical editor so you can review the files about to be uploaded and make any corrections you wish.

For example, to create a new review and add all changes in the current directory and below, plus the file foo.txt, you would use:

```
ccollab addchanges new . foo.txt
```

To upload modified files from the current working directory and all subdirectories to review 123:

```
ccollab addchanges 123 .
```

To upload file foo.txt and modified files from c:\dev\project into a brand new review:

```
ccollab addchanges new foo.txt c:\dev\project
```

The '--diffbranch' option allows local changes to be uploaded with a predecessor version in another branch as the previous version.

For example, to create a new review and add all changes in the current directory and below, and compare them to the predecessor versions in the \main branch, you would use:

© 2003-2011 SmartBear Software

```
ccollab addchanges --diffbranch main new .
```

You can also use the special branch name 'pre' to refer to the predecessor branch. For example, to add the local changes to hello.c compared to the most recent predecessor version in the previous branch to review 45, you would use:

```
ccollab addchanges --diffbranch pre 45 hello.c
```

5.6.2.2 addchangelist (for ClearCase)

```
Usage: ccollab [global-options] addchangelist <review> <changelist>
[<changelist> ...]
```

`addchangelist` - Attaches an atomic changelist to a review

Command Options

Option	Required?	Description
<review>	Y	Must be either an integer review-id, 'new', 'ask', or 'last'
<changelist> [<changelist> ...]	Y	SCM-specific ID of an atomic set of changes

Uploads Activities or Activities from ClearCase to the PeerReview Complete server.

The first argument is the review specifier, subsequent arguments are the IDs of the Activities or Activities to upload.

For example, to upload Activities add_new_scm_support and add_extended_version_parsing to a new review:

```
ccollab addchangelist new add_new_scm_support add_extended_version_parsing
```

For example, to upload Activities AnActivity and AnotherActivity to review 111:

© 2003-2011 SmartBear Software

ccollab addchangelist 111 AnActivity AnotherActivity

5.6.2.3 addversions (for ClearCase)

```
Usage: ccollab [global-options] addversions [--upload-comment <value>]
<review> [<version-spec> [<version-spec> ...]]
```

`addversions` - Attaches any 2 given versions to a review

Command Options

Option	Required?	Description
--upload-comment <value>	N	Comment used to upload files (default is "Local changes")
<review>	Y	Must be either an integer review-id, 'new', 'ask', or 'last'
<version-spec> [<version-spec> ...]	N	Versions to be reviewed are given by <path> <version> [<previous version>], where <path> is the filename or server path of the file, <version> is the version to be reviewed, and <previous version> is an optional version that <version> should be diff'ed against. If <previous version> is not specified, the predecessor version of <version> will be used. If no <version-spec> is given on the command line, then one or more version specs will be read from stdin, one <version-spec> per line as described above. Note that arguments containing spaces must be quoted to be parsed correctly

Uploads specific versions of a file or files to a review. A version-spec is given by a filename followed by a version number, and optionally, a previous version number:

© 2003-2011 SmartBear Software

file-name version-number [previous-version-number]

If no previous version number is explicity given, diffs will be generated with the predecessor version as determined from the SCM system.

Adding versions for a single file can be done with command line arguments given after the review number. To upload versions for multiple files, the file and versions may be submitted on lines read from standard input. If the path and versions are not given on the command line, the client will expect to read them from stdin. Each line read should be in the format shown above. Some example command lines for reading versions from stdin:

```
ccollab addversions 86753
ccollab addversions last < versionlist.txt
cat versionlist.txt | ccollab addversions new
```

When supplying the version list in the command line client editor or an input file, the keyword 'local' can be used to denote the version corresponding to the local version of the file. The 'local' keyword can only be used for the first version argument, not the previous version.

Examples of specifying versions on the command line for ClearCase:

```
ccollab addversions new hello.c \main\mydev\6 \main\8
ccollab addversions last Main.java /main/dev/31
```

To compare the local version of the ClearCase file to a previous version in an edit list or input list, use 'local' as the initial version. If the file is checked out as '/main/CHECKEDOUT', then 'local' is equivalent to using '/main/CHECKEDOUT' as the initial version. Otherwise 'local' would refer to

© 2003-2011 SmartBear Software

the latest version of the file. For example, if the local file hello.c is
at version '/main/9' and has predecessor '/main/8' then all of the following
lines would be equivalent:

hello.c /main/9
hello.c local
hello.c /main/9 /main/8
hello.c local /main/8

You can also reference a LATEST version, e.g.:

ccollab addversions new hello.c \main\ga_1.0\6 \main\LATEST

To upload a version with no predecessor version, use '\main\0':

ccollab addversions new hello.c \main\ga_1.0\6 \main\0

5.6.2.4 addactivity

```
Usage: ccollab [global-options] addactivity [--diffintegration --upload-
comment <value>] <review> <activity-name> [<activity-name> ...]
```

`addactivity` - Attaches file versions in a ClearCase activity to a review

Command Options

Option	Required?	Description
--diffintegration	N	Use LATEST version from default integration stream, if available, as the predecessor version
--upload-comment <value>	N	Comment used to upload files (default is "Local changes")

© 2003-2011 SmartBear Software

Option	Required?	Description
<review>	Y	Must be either an integer review-id, 'new', 'ask', or 'last'
<activity-name> [<activity-name> ...]	Y	Specify one or more ClearCase activities by name, or use 'all' to include all activities in the current view

Attaches file versions from one or more activities to a review. For each file in the activity, the latest version in the activity and the predecessor of the earliest version in the activity are uploaded to the review. You can specify multiple activity names on the command line. You can also use the word 'all' as the activity name to include all activities in the current view. If multiple activities are given, the changeset uploaded is the union of all changes in each of the activities.

You can use the --diffintegration option to include the default integration stream in determining version content for review. For the most recent version, this option will scan the versions of each file in the given activity as well as the versions of those files in any associated rebase activities, to determine the most recent version. For the previous version for each file in the activity, this option will use the LATEST version in the default integration branch (if available) as the previous version.

5.6.2.5 commit (for ClearCase)

```
Usage: ccollab [global-options] commit [--comment <value> --dismiss-only --
force] <review>
```

`commit` - Commit changes in the review

Command Options

Option	Required?	Description
--comment <value>	N	Comment for reviewed changes
--dismiss-only	N	Just dismiss the Action Item
--force	N	Ignore potential problems
<review>	Y	Must be either an integer review-id, 'ask', or 'last'

© 2003-2011 SmartBear Software

Commit the changes from a pre-commit review to sourcecontrol. Be sure to include a relevant comment.

5.7 IBM Rational Synergy Integration

This section describes PeerReview Complete integration with Rational Synergy:

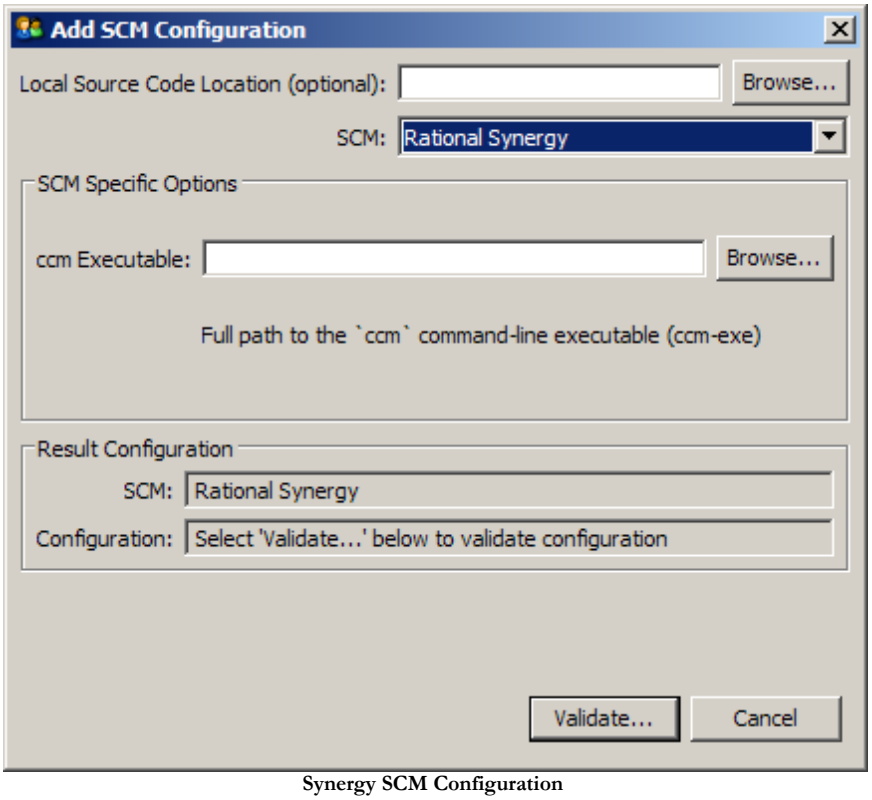
GUI Client 336
The GUI Client can upload Tasks into PeerReview Complete. You can upload Pending 337 or Completed 338 Tasks

Command Line Client 339
The Command Line Client can upload Tasks into Command Line Client.

Supported Versions
Our integration uses your own Rational Synergy command-line client (ccm) to communicate with the server. The integration was developed against the Rational Synergy 7.1 toolset, and should work with the latest 6.x and 7.x releases.

5.7.1 GUI Client

Rational Synergy-specific Options
The SCM Configuration dialog 191 has several Rational Synergy-specific options. These can be set as necessary to override Rational Synergy options derived from the environment.

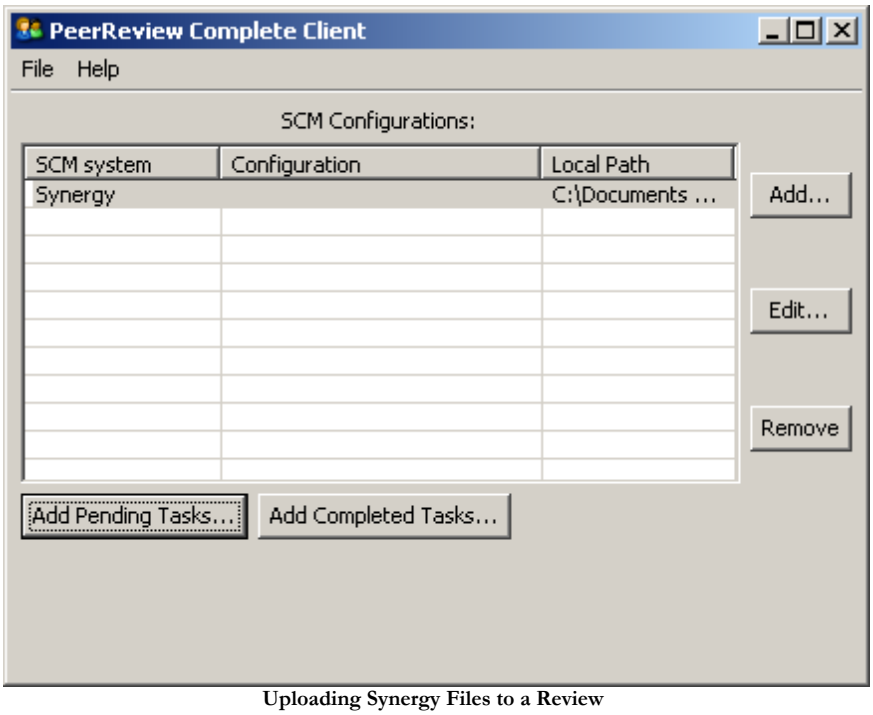


Synergy SCM Configuration

© 2003-2011 SmartBear Software

Uploading files to a Review

Selecting a Rational Synergy SCM Configuration in the GUI Client main screen 190 causes several Add to Review buttons to appear. The *Add Pending Tasks...* 337 button uploads pending changelists. The *Add Submitted Changelists...* 370 button uploads submitted changelists.

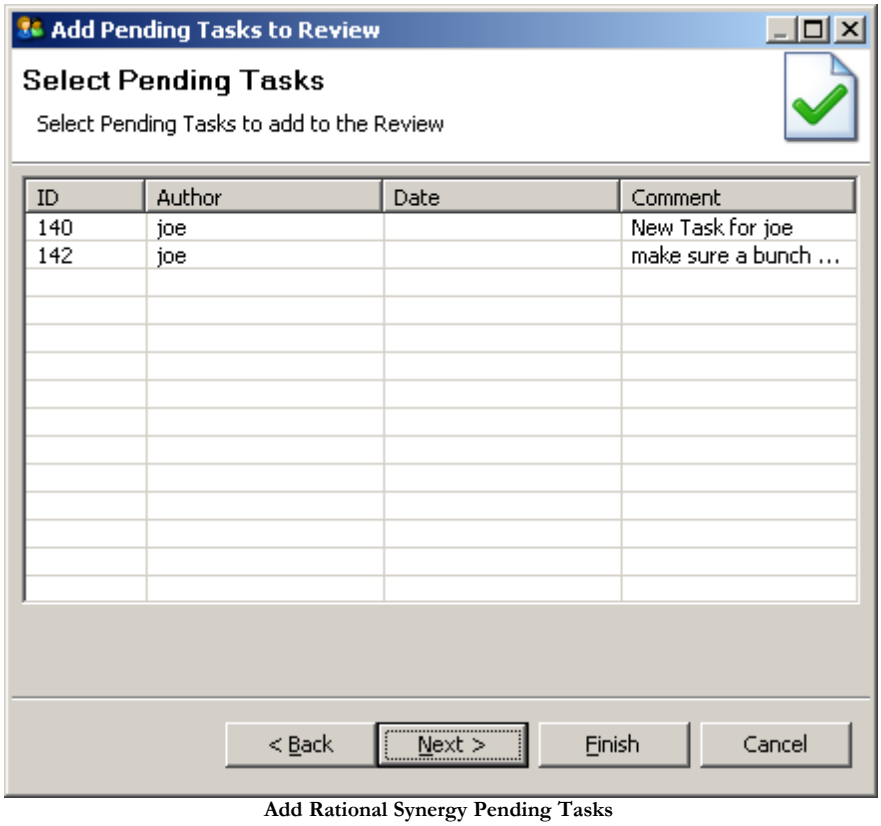


Uploading Synergy Files to a Review

Add Pending Tasks

Press the *Add Pending Tasks...* button to upload the files in a pending Task to the PeerReview Complete Server for review.

© 2003-2011 SmartBear Software

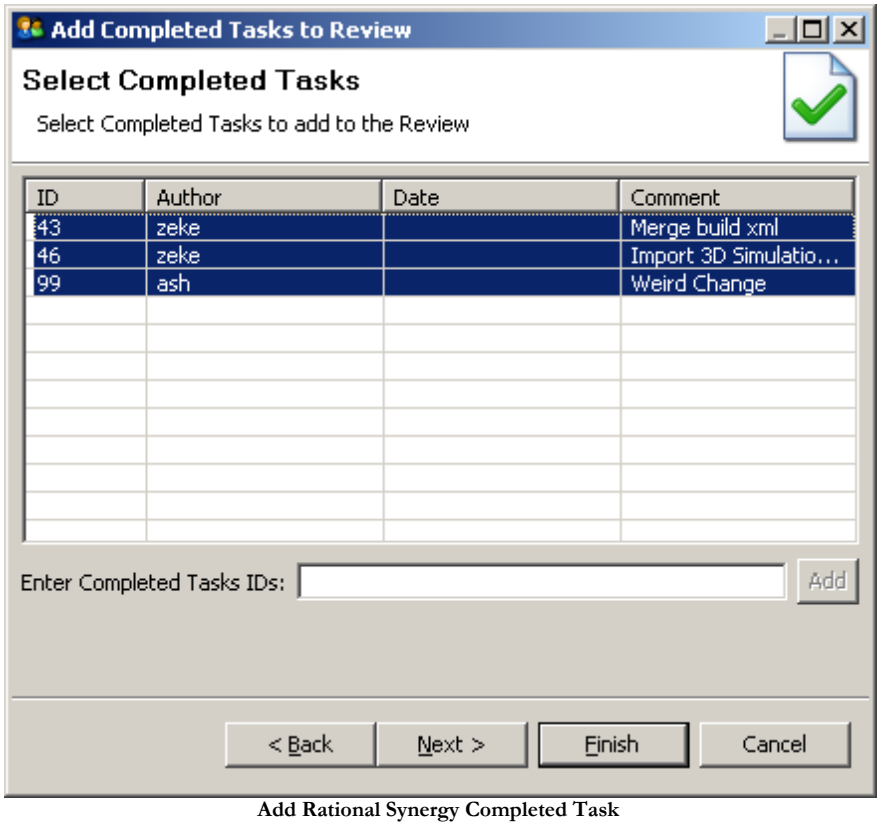


Add Rational Synergy Pending Tasks

Add Rational Synergy Completed Tasks

Press the *Add Completed Tasks...* button to upload the files in a completed Task to the PeerReview Complete Server for review.

© 2003-2011 SmartBear Software



Add Rational Synergy Completed Task

5.7.2 Command Line Client

Commands recommended for Rational Synergy

ccollab addchangelist 340 - Attaches an atomic changelist to a review

ccollab addversions 341 - Attaches any 2 given versions to a review

The addchangelist 375 command will upload Rational Synergy tasks into PeerReview Complete. You can upload tasks either before or after they are completed.

Configuration

The Command Line Client can may need some configuration. Trytesting your configuration 197 to verify the configuration is detected correctly.

If the Command Line Client is unable to detect your Rational Synergy configuration or you want to override the detected settings, you can manually specify Rational Synergy settings using global options 207.

To manually configure the Command Line Client to use Rational Synergy, execute the following command:

`ccollab set` 215 `scm` 205 `synergy`

Rational Synergy-specific Options

© 2003-2011 SmartBear Software

Option	Description
--ccm-exe <value>	Full path to the `ccm` command-line executable
--ccm-user <value>	User Name to use when starting a Rational Synergy session
--ccm-passwd <value>	Password to use when starting a Rational Synergy session
--ccm-engine-host <value>	Host the Rational Synergy Engine will run on
--ccm-database-path <value>	Path of the Rational Synergy database to connect with
--ccm-local-database-path <value>	Path of the local Rational Synergy database, typically c:/temp/ccm or /tmp/ccm
--ccm-remote-client	Start Rational Synergy sessions as a Remote Client

5.7.2.1 addchangelist (for Rational Synergy)

```
Usage: ccollab [global-options] addchangelist <review> <changelist>
[<changelist> ...]
```

`addchangelist` - Attaches an atomic changelist to a review

Command Options

Option	Required?	Description
<review>	Y	Must be either an integer review-id, 'new', 'ask', or 'last'
<changelist> [<changelist> ...]	Y	SCM-specific ID of an atomic set of changes

Uploads Pending Tasks or Completed Tasks from Rational Synergy to the PeerReview Complete server.

The first argument is the review specifier, subsequent arguments are the IDs of the Pending Tasks or Completed Tasks to upload.

© 2003-2011 SmartBear Software

For example, to upload Pending Tasks 4321 and 7568 to a new review:

```
ccollab addchangelist new 4321 7568
```

For example, to upload Completed Tasks 5432 and 12654 to review 111:

```
ccollab addchangelist 111 5432 12654
```

5.7.2.2 addversions (for Rational Synergy)

```
Usage: ccollab [global-options] addversions [--upload-comment <value>]
<review> [<version-spec> [<version-spec> ...]]
```

`addversions` - Attaches any 2 given versions to a review

Command Options

Option	Required?	Description
--upload-comment <value>	N	Comment used to upload files (default is "Local changes")
<review>	Y	Must be either an integer review-id, 'new', 'ask', or 'last'

© 2003-2011 SmartBear Software

Option	Required?	Description
<version-spec> [<version-spec> ...]	N	Versions to be reviewed are given by <path> <version> [<previous version>], where <path> is the filename or server path of the file, <version> is the version to be reviewed, and <previous version> is an optional version that <version> should be diff'ed against. If <previous version> is not specified, the predecessor version of <version> will be used. If no <version-spec> is given on the command line, then one or more version specs will be read from stdin, one <version-spec> per line as described above. Note that arguments containing spaces must be quoted to be parsed correctly

Uploads specific versions of a file or files to a review. A version-spec is given by a filename followed by a version number, and optionally, a previous version number:

file-name version-number [previous-version-number]

If no previous version number is explicity given, diffs will be generated with the predecessor version as determined from the SCM system.

Adding versions for a single file can be done with command line arguments given after the review number. To upload versions for multiple files, the file and versions may be submitted on lines read from standard input. If the path and versions are not given on the command line, the client will expect to read them from stdin. Each line read should be in the format shown above. Some example command lines for reading versions from stdin:

© 2003-2011 SmartBear Software

```
ccollab addversions 86753
ccollab addversions last < versionlist.txt
cat versionlist.txt | ccollab addversions new
```

When supplying the version list in the command line client editor or an input file, the keyword 'local' can be used to denote the version corresponding to the local version of the file. The 'local' keyword can only be used for the first version argument, not the previous version.

An example of specifying versions on the command line:

```
ccollab addversions review hello.c:csrc:1 3.2.5 3.2
```

5.8 Mercurial Integration

This section describes PeerReview Complete integration with Mercurial:

GUI Client 344
The GUI Client can upload arbitrary Mercurial diffs. The GUI Client can also upload local changes to files that are managed by Mercurial.

Command Line Client 346
The Command Line Client can upload arbitrary Mercurial diffs and uncommitted changes to files that are managed by Mercurial.

Supported Versions
Our integration uses your own installed Mercurial command line client executable to generate differences for review. We require hg v1.0 or later, as we use the -U flag to provide full context lines of differences so that the uploaded versions can contain the full file content of the previous and current file versions.

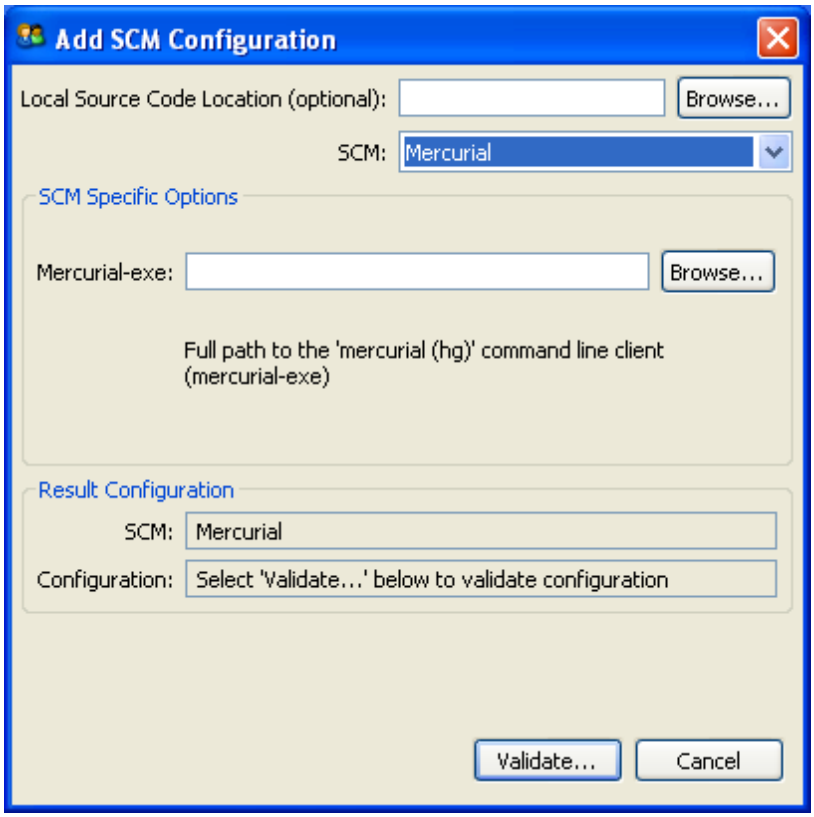
Because we use client applications already present on your computer, we support all protocols, authentications, proxies, and other client configuration options you are currently using.

© 2003-2011 SmartBear Software

5.8.1 GUI Client

Mercurial-specific Options

The SCM Configuration dialog 191 has one Mercurial-specific option. The full path to the Mercurial command line executable on your local file system should be specified if it is not already on your system path.

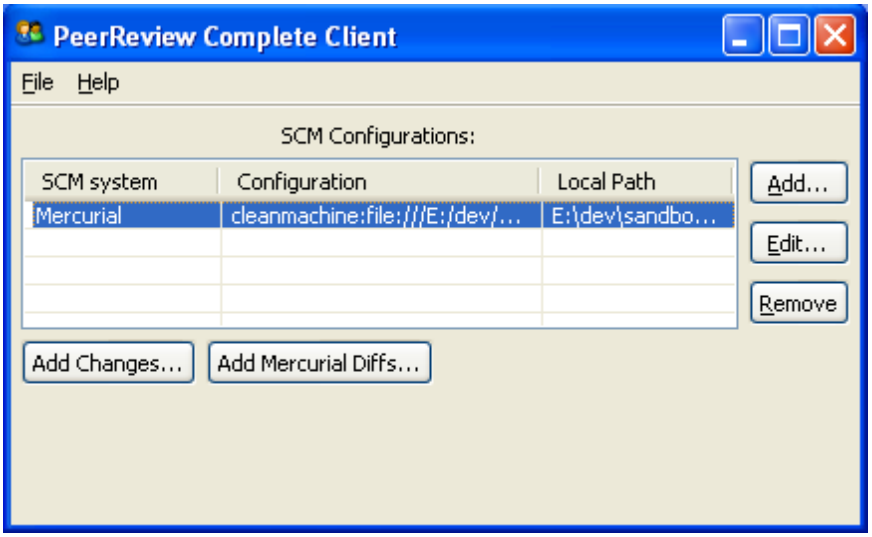


Configure client to work with Mercurial

Uploading files to a Review

Selecting a Mercurial SCM Configuration in the GUI Client main screen 190 causes *Add to Review buttons* to appear.

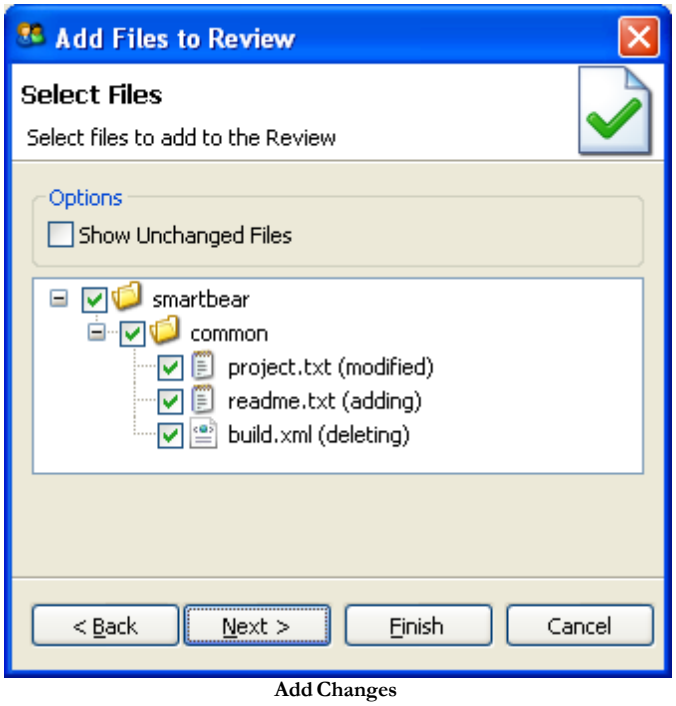
© 2003-2011 SmartBear Software



Uploading Mercurial files to a review

Add Changes

The *Add Changes...* button will be disabled if you have not specified a working copy in the Local Path field of the SCM Configuration dialog 191. Selecting *Add Changes...* allows you to upload your local modifications. These are the modifications that would be committed if you typed 'hg commit' from a command line.

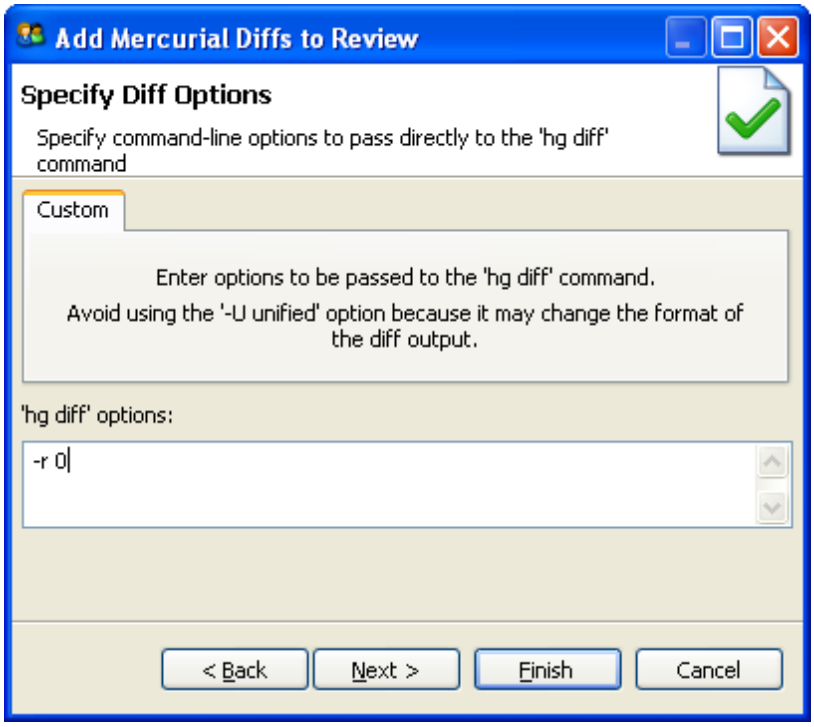


Add Changes

Add Mercurial Diffs

Press the *Add Mercurial Diffs...* button to upload arbitrary Mercurial diffs to the PeerReview Complete Server for review. Note that if you did not specify a Local Source Code Location in the SCM Configuration dialog 191, then you will need to provide absolute paths in your diff options.

© 2003-2011 SmartBear Software


Add Mercurial Diffs to Review
Specify Diff Options
Specify command-line options to pass directly to the 'hg diff' command
Custom
Enter options to be passed to the 'hg diff' command.
Avoid using the '-U unified' option because it may change the format of the diff output.
'hg diff' options:
-r 0
< Back
Next >
Finish
Cancel

Upload arbitrary Mercurial diffs

5.8.2 Command Line Client

Commands recommended for Mercurial

ccollab addchanges 346 - Attaches locally-modified files to a review

ccollab addhgdiffs 223 - Uploads diffs generated by hg diff command

Configuration

No extra configuration is required to use Mercurial with the Command Line Client

5.8.2.1 addchanges (for Mercurial)

```
Usage: ccollab [global-options] addchanges [--diffbranch <value> --upload-
comment <value>] <review> <file-spec> [<file-spec> ...]
```

`addchanges` - Attaches locally-modified files to a review

Command Options

© 2003-2011 SmartBear Software

Option	Required?	Description
--diffbranch <value>	N	(ClearCase only) specify either branch-name or "predecessor" to determine the earlier version for upload
--upload-comment <value>	N	Comment used to upload files (default is "Local changes")
<review>	Y	Must be either an integer review-id, 'new', 'ask', or 'last'
<file-spec> [<file-spec> ...]	Y	Local files or directories to scan for modified files

Uploads locally modified files controlled by Mercurial to the PeerReview Complete server.

The first argument is the review specifier, then any number of files or directories can be given. Files are selected as-is; directories are scanned recursively for locally-modified files, which might include edits, additions, deletions, branches, integrations, moves, copies, and so on.

After the scan is complete, a file list is presented in a graphical editor so you can review the files about to be uploaded and make any corrections you wish.

For example, to create a new review and add all changes in the current directory and below, plus the file foo.txt, you would use:

```
ccollab addchanges new . foo.txt
```

To upload modified files from the current working directory and all subdirectories to review 123:

```
ccollab addchanges 123 .
```

To upload file foo.txt and modified files from c:\dev\project into a brand new review:

```
ccollab addchanges new foo.txt c:\dev\project
```

© 2003-2011 SmartBear Software

5.8.2.2 addhgdiffs

```
Usage: ccollab [global-options] addhgdiffs [--upload-comment <value>]
<review> [<user-diff-arg> [<user-diff-arg> ...]]
```

`addhgdiffs` - Uploads diffs generated by hg diff command

Command Options

Option	Required?	Description
--upload-comment <value>	N	Comment used to upload files (defaults to command-line arguments)
<review>	Y	Must be either an integer review-id, 'new', 'ask', or 'last'
<user-diff-arg> [<user-diff-arg> ...]	N	Command-line arguments to pass directly to the diff command

Upload differences between arbitrary versions of files in Mercurial using 'hg diff'.

Do not use diff arguments that affect the diff output such as -U unified. The PeerReview Complete command-line client will automatically select an output format that ensures you will get all the data you need on the server.

For example, to upload all changes between revisions 4 and 8:

ccollab addhgdiffs review -r 4 -r 8

To upload all changes in your local working directory:

ccollab addhgdiffs review .

© 2003-2011 SmartBear Software

5.9 Microsoft Team Foundation Server Integration

This section describes PeerReview Complete integration with Team Foundation Server.

GUI Client 349
The GUI Client can upload local changes to files 351 in a Team Foundation Server Workspace. The GUI Client can also upload the files in a Shelveset 351 or Changeset 351.

Command Line Client 352
The Command Line Client can upload local changes to files 353 controlled by Team Foundation Server before they are checked in. The Command Line Client can also uploadthe files in a Shelveset 354 or Changeset 354.

Supported Versions
Team Foundation Server versions 2005 through 2010 are supported bythe Command Line Client 352 and the GUI Client 349.

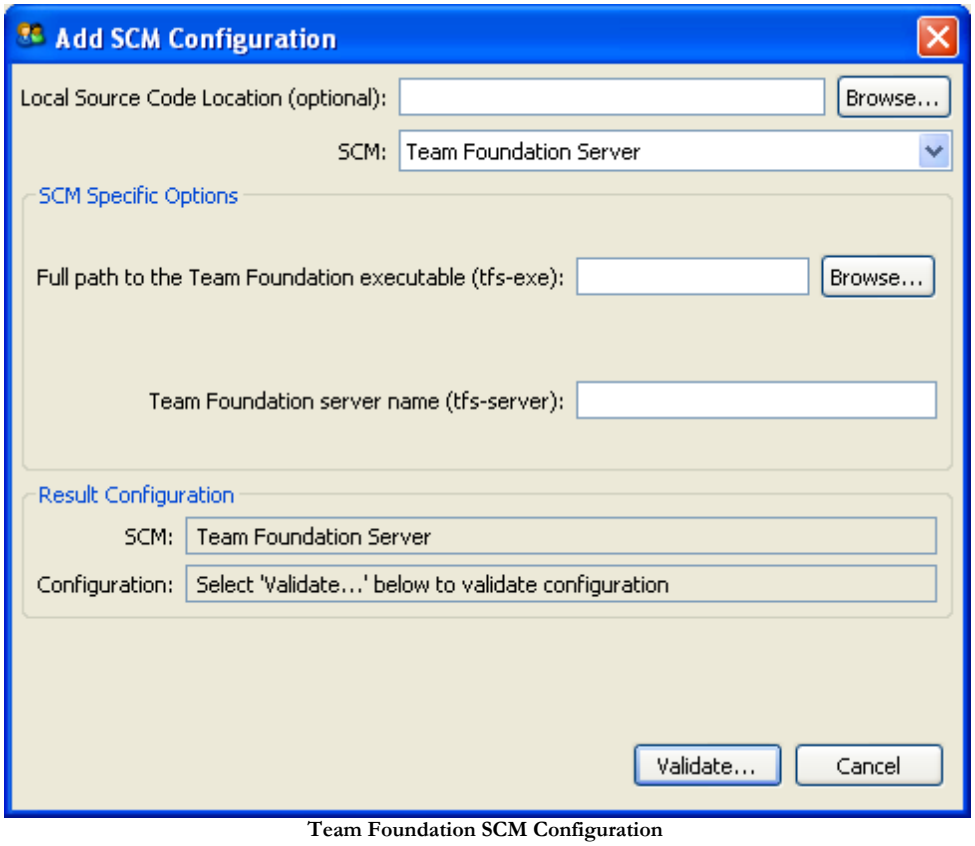
Our client integrations use the installed TF command line client executable (tf .exe) to communicate with Team Foundation Server. Because we use client applications already present on your computer, we support all protocols, authentications, proxies, and other client configuration options you are currently using.

The Team Foundation Server integration will not work with non-English installations of Visual Studio .NET. Regional settings for other locales are supported, but installing a non-English Visual Studio prevents correct parsing of the TF command line output.

5.9.1 GUI Client

Team Foundation specific Options
The SCM Configuration dialog 191 has several Team Foundation specific options. If you have no Team Foundation Workspace on your computer, you can still review Team Foundation Shelvesets or Changesets by specifying the Team Foundation server name option.

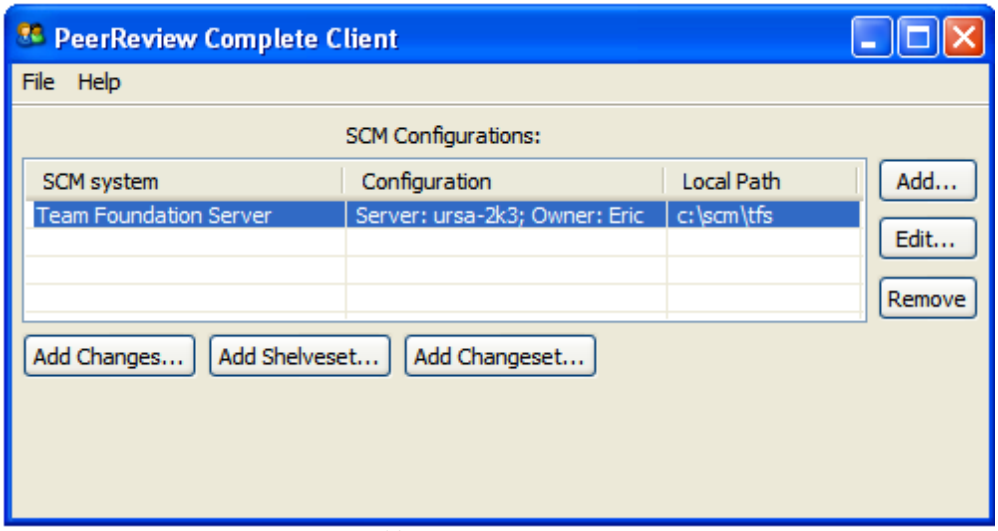
© 2003-2011 SmartBear Software



Team Foundation SCM Configuration

Uploading files to a Review

Selecting an Team Foundation SCM Configuration in the GUI Client main screen 190 causes three Add to Review buttons to appear. The *Add Changes...* 351 button uploads modified files in a Workspace. The *Add Shelvesets...* 351 button uploads the files in Shelvesets. The *Add Changesets...* 351 button uploads the files in committed Team Foundation Changesets.

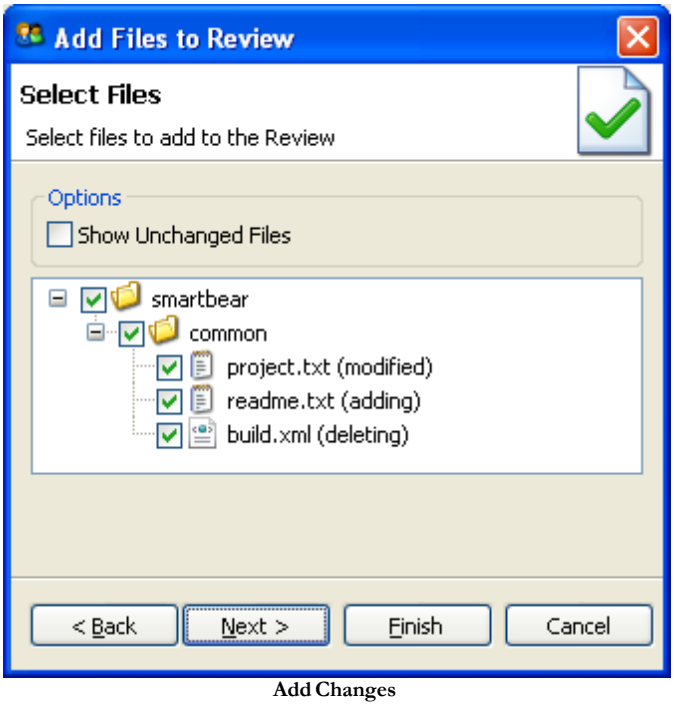


Uploading Team Foundation files to a Review

© 2003-2011 SmartBear Software

Add Changes

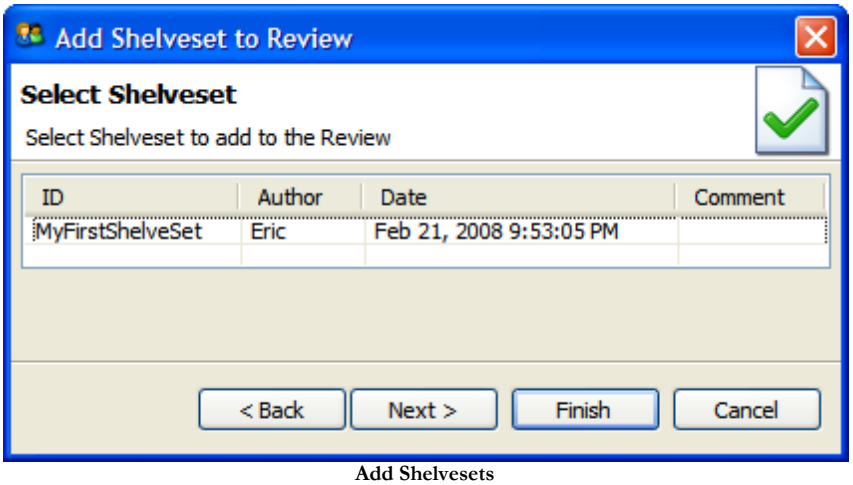
Press the *Add Changes...* button to upload modified files in a Workspace to the PeerReview Complete Server for review.



Add Changes

Add Shelvesets

Press the *Add Shelvesets...* button to upload the files in a Shelveset to the PeerReview Complete Server for review. See the *Add Changesets...* button if you want to add a Shelveset to the review that's not in your list of shelvesets.

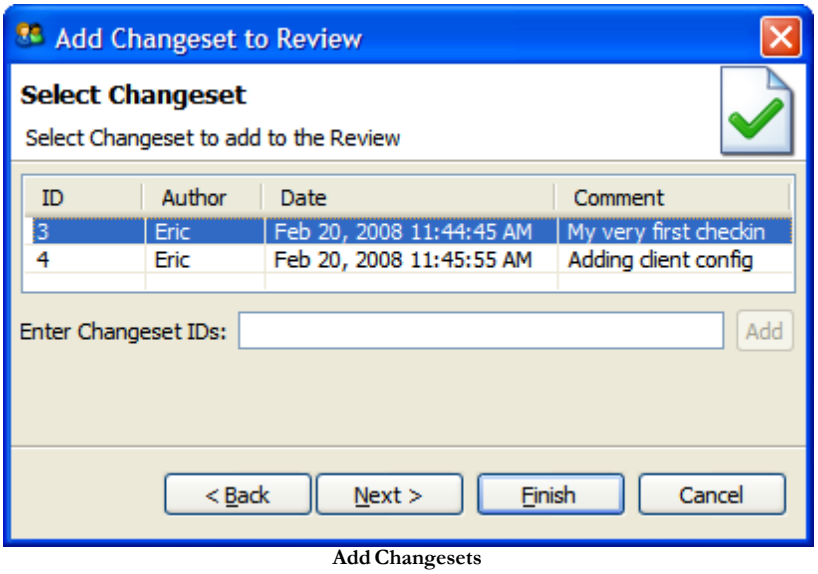


Add Shelvesets

Add Changesets

Press the *Add Changesets...* button to upload the files in a Team Foundation Changeset to the PeerReview Complete Server for review. You can also add a named Shelveset to the review in this dialog by entering a shelveset name instead of a changeset number.

© 2003-2011 SmartBear Software



Add Changesets

5.9.2 Command Line Client

Commands recommended for Team Foundation Server

ccollab addchanges 353 - Attaches locally-modified files to a review

ccollab addchangelist 354 - Attaches an atomic changelist to a review

ccollab commit 355 - Commit changes in the review

The addchanges 353 command will upload local changes to files controlled by Team Foundation before they are checked in.

The addchangelist 354 command will upload a the files in a Shelveset or Changeset.

Configuration

In most cases, the Command Line Client can automatically detect your Team Foundation configuration. Try testing your configuration 197 to verify the configuration is detected correctly.

If the Command Line Client is unable to detect your Team Foundation configuration or you want to override the detected settings, you can manually specify Team Foundation settings using global options 204.

To manually configure the Command Line Client to use Team Foundation, execute the following command:

```
ccollab set 215 scm 205 tfs
```

Team Foundation Server-specific Options

© 2003-2011 SmartBear Software

Option	Description
--tfs-collection <value>	Team Foundation collection name (2010 and later only)
--tfs-server <value>	Team Foundation server name (2005 and 2008)
--tfs-exe <value>	Full path to the Team Foundation `tf` command-line client
--tfs-user <value>	Team Foundation alternate user name (atypical)
--tfs-passwd <value>	Team Foundation alternate user name password

5.9.2.1 addchanges (for Team Foundation Server)

```
Usage: ccollab [global-options] addchanges [--diffbranch <value> --upload-
comment <value>] <review> <file-spec> [<file-spec> ...]
```

`addchanges` - Attaches locally-modified files to a review

Command Options

Option	Required?	Description
--diffbranch <value>	N	(ClearCase only) specify either branch-name or "predecessor" to determine the earlier version for upload
--upload-comment <value>	N	Comment used to upload files (default is "Local changes")
<review>	Y	Must be either an integer review-id, 'new', 'ask', or 'last'
<file-spec> [<file-spec> ...]	Y	Local files or directories to scan for modified files

Uploads locally modified files controlled by Team Foundation Server to the PeerReview Complete server.

© 2003-2011 SmartBear Software

The first argument is the review specifier, then any number of files or directories can be given. Files are selected as-is; directories are scanned recursively for locally-modified files, which might include edits, additions, deletions, branches, integrations, moves, copies, and so on.

After the scan is complete, a file list is presented in a graphical editor so you can review the files about to be uploaded and make any corrections you wish.

For example, to create a new review and add all changes in the current directory and below, plus the file foo.txt, you would use:

ccollab addchanges new . foo.txt

To upload modified files from the current working directory and all subdirectories to review 123:

ccollab addchanges 123 .

To upload file foo.txt and modified files from c:\dev\project into a brand new review:

ccollab addchanges new foo.txt c:\dev\project

5.9.2.2 addchangelist (for Team Foundation Server)

```
Usage: ccollab [global-options] addchangelist <review> <changelist>
[<changelist> ...]
```

`addchangelist` - Attaches an atomic changelist to a review

Command Options

Option	Required?	Description
<review>	Y	Must be either an integer review-id, 'new', 'ask', or 'last'
<changelist> [<changelist> ...]	Y	SCM-specific ID of an atomic set of changes

Uploads Shelvesets or Changesets from Team Foundation Server to the PeerReview Complete server.

© 2003-2011 SmartBear Software

The first argument is the review specifier, subsequent arguments are the IDs of the Shelvesets or Changesets to upload.

Note that the Changesets are searched first - if a Changeset is found, it will be added to the review; to avoid any naming conflicts, always use at least one non-numeric character in your Shelveset names.

For example, to upload Shelvesets MyShelveset and todays_work to a new review:

```
ccollab addchangelist new MyShelveset todays_work
```

For example, to upload Changesets C3 and C12654 to review 111:

```
ccollab addchangelist 111 3 12654
```

5.9.2.3 commit (for Team Foundation Server)

```
Usage: ccollab [global-options] commit [--comment <value> --dismiss-only --
force] <review>
```

`commit` - Commit changes in the review

Command Options

Option	Required?	Description
--comment <value>	N	Comment for reviewed changes
--dismiss-only	N	Just dismiss the Action Item
--force	N	Ignore potential problems
<review>	Y	Must be either an integer review-id, 'ask', or 'last'

Commit the changes from a pre-commit review to sourcecontrol. Be sure to include a relevant comment.

© 2003-2011 SmartBear Software

5.10 Microsoft Visual SourceSafe Integration

This section describes PeerReview Complete integration with Visual SourceSafe.

Command Line Client 356
Our Visual SourceSafe support is very limited in functionality. The Command Line Client supports creating reviews from SourceSafe file differences.

The Command Line Client installs with a shell executable, `ssdiff.exe`, that calls the Visual SourceSafe command line client (`ss.exe`). This executable has a dependency on redistributable MFC DLL's. If ssdiff.exe is unable to execute due to a missing DLL file, download and install MFC redistributable DLL's for ssdiff.exe to your Windows system directory. Any differences that `ss.exe` can generate can be added to reviews using Command Line Client. Typically, though, this is a single file at a time.

Supported Versions
Our integration supports Microsoft Visual SourceSafe v6.0 and higher.

Because we use client applications already present on your computer, we support all protocols, authentications, proxies, and other client configuration options you are currently using.

5.10.1 Command Line Client

Command recommended for Visual SourceSafe

ccollab addvssdiffs 226 - Uploads diffs generated from 'ss diff' command

Configuration
To configure the Command Line Client to use SourceSafe, execute the following command:

```
ccollab set 215 scm 205 vss
```

You can tailor the Command Line Client for your SourceSafe configuration using global options 211. These can be specified on the command line, or set as persistent values using `ccollab set` 215.

Visual SourceSafe-specific Options

Option	Description
--vss-passwd <value>	SourceSafe password
--vss-exe <value>	Full path to the SourceSafe `ss` command-line client
--vss-user <value>	SourceSafe user name

Example:

```
ccollab set 215 vss-exe "C:\Program Files\Microsoft Visual Studio
```

© 2003-2011 SmartBear Software

```
\Common\VSS\win32\ss.exe"
```

5.10.1.1 addvssdiffs

```
Usage: ccollab [global-options] addvssdiffs [--upload-comment <value>]
<review> [<user-diff-arg> [<user-diff-arg> ...]]
```

`addvssdiffs` - Uploads diffs generated from 'ss diff' command

Command Options

Option	Required?	Description
--upload-comment <value>	N	Comment used to upload files (defaults to command-line arguments)
<review>	Y	Must be either an integer review-id, 'new', 'ask', or 'last'
<user-diff-arg> [<user-diff-arg> ...]	N	Command-line arguments to pass directly to the diff command

Upload differences between arbitrary versions of files in Visual SourceSafe using 'ss diff'.

Do not use diff arguments that affect the diff output such as -o, -b, -h, -?, -du<width> where width is less than 128, or -dx<context> where context is less than 10. The PeerReview Complete command-line client will automatically select an output format that ensures you will get all the data you need on the server.

Examples:

ccollab addvssdiffs 698 HelloWorld.java -yusername,password

ccollab addvssdiffs new HelloWorld.java -v2

ccollab addvssdiffs 698 HelloWorld.java -v4~1

ccollab addvssdiffs 698 $/Project/Status.java c:\project\Status.java

© 2003-2011 SmartBear Software

5.11 MKS Source Integrity Integration

PeerReview Complete provides the following clients to integrate with MKS Source Integrity.

GUI Client 358
The GUI Client integrates with MKS Source Integrity. It can find and upload changes by pending or committed MKS Change Packages, by specific file version number, or by modified working files in local sandboxes.

Command Line Client 361
The Command Line Client integrates with MKS Source Integrity. It can find and upload changes by pending or committed MKS Change Packages, or by modified working files in local sandboxes.

Supported Versions
Our integration uses the Java client API for MKS Source Integrity 2007. This API communicates with the installed MKS client on your system. It supports the MKS 2007 client, and MKS 2006 clients with the latest fixpacks.

Because we use client applications already present on your computer, we support all protocols, authentications, proxies, and other client configuration options you are currently using.

A Note about MKS Change Packages
Uploading change packages for review generates file differences based on Multiple Version Changelists 458.

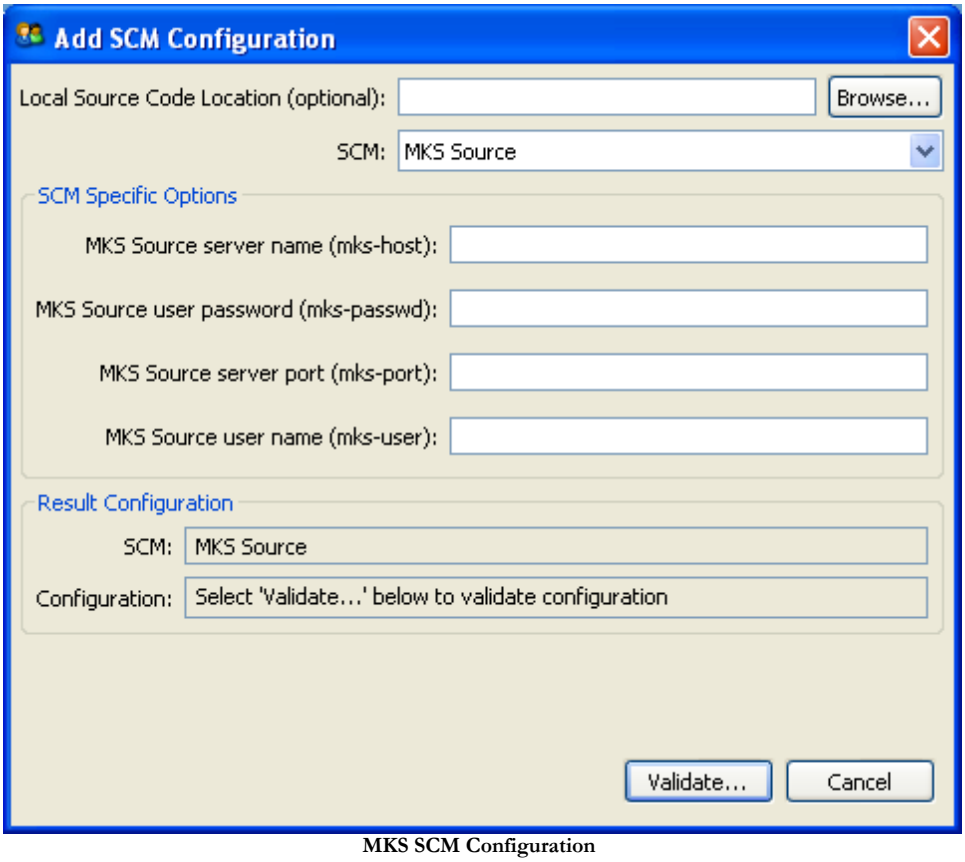
Linux Installation
Under Linux, the Java client API for MKS Source Integrity requires these environment variables to be set for our clients to function correctly:

```
     export PATH=[integrity-client-install-dir]/bin:$PATH
     export LD_LIBRARY_PATH=[integrity-client-install-dir]/lib/
linux:$LD_LIBRARY_PATH
```

5.11.1 GUI Client

MKS-specific Options
The SCM Configuration dialog 191 has several MKS-specific options. If your MKS Source or Integrity client automatically connects to the MKS server using default local settings for some or all of these options, you need not specify them.

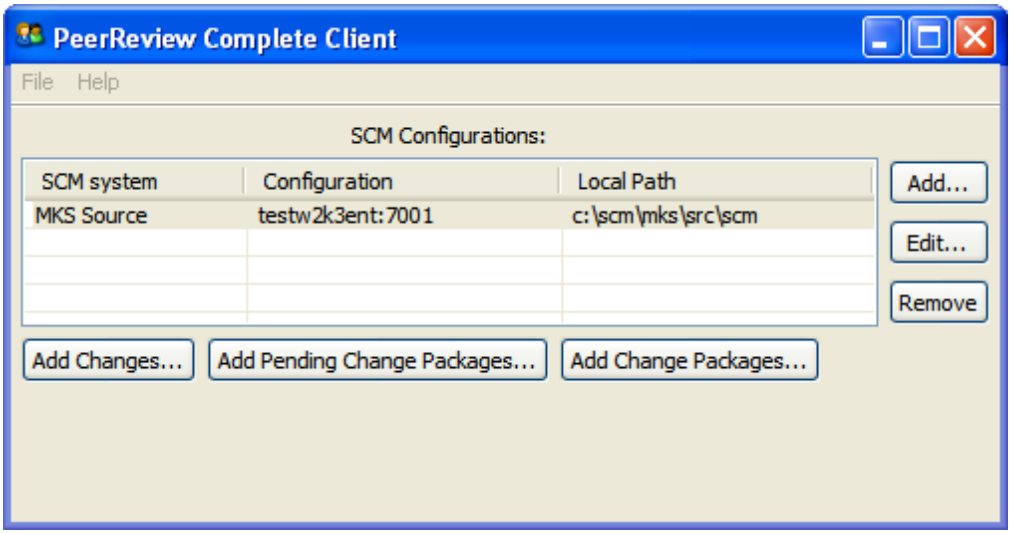
© 2003-2011 SmartBear Software



MKS SCM Configuration

Uploading files to a Review

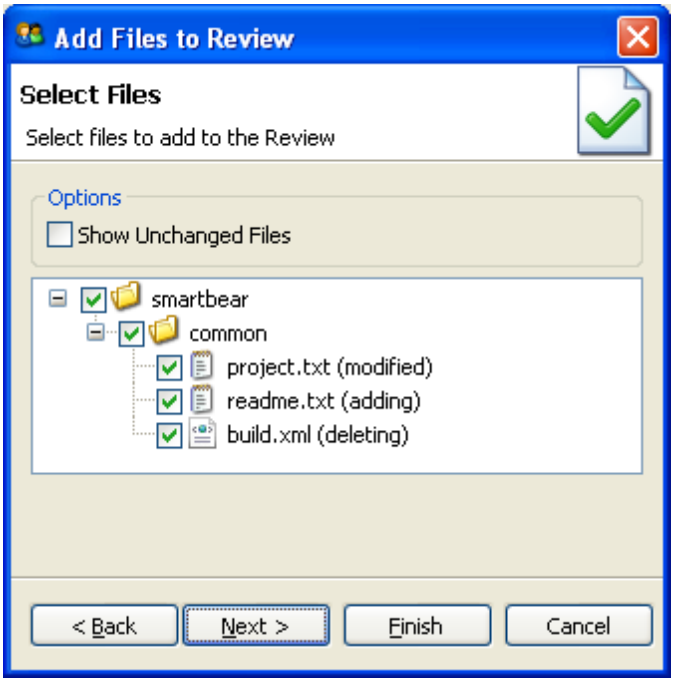
Selecting a Perforce SCM Configuration in the GUI Clientmain screen 190 causes several Add to Review buttons to appear. The *Add Pending Change Packages...* 360 button uploads pending changelists. The *Add Change Packages...* 360 button uploads submitted changelists.



© 2003-2011 SmartBear Software

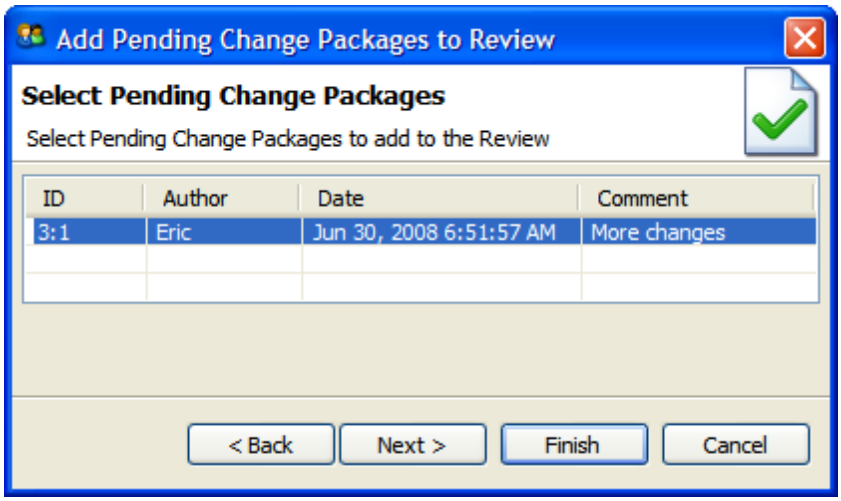
Add Changes

Press the *Add Changes...* button to upload modified files from local MKS sandboxes to the PeerReview Complete Server for review.



Add Pending MKS Change Packages

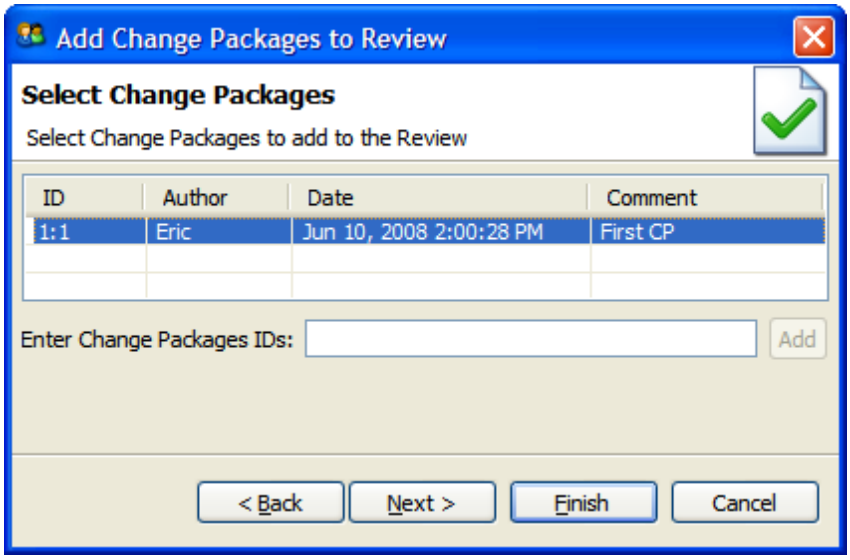
Press the *Add Pending Change Packages...* button to upload the files in a pending MKS Change Package to the PeerReview Complete Server for review.



Add Submitted MKS Change Packages

Press the *Add Change Packages...* button to upload the files in an already committed MKS Change Package to the PeerReview Complete Server for review.

© 2003-2011 SmartBear Software



5.11.2 Command Line Client

Commands recommended for MKS Source

ccollab addchanges 362 - Attaches locally-modified files to a review

ccollab addchangelist 363 - Attaches an atomic changelist to a review

ccollab addversions 364 - Attaches any 2 given versions to a review

Configuration

In most cases, the Command Line Client can automatically detect your MKS configuration. Try testing your configuration 197 to verify the configuration is detected correctly.

If the Command Line Client is unable to detect your MKS configuration or you want to override the detected settings, you can manually specify MKS settings using global options 204.

To manually configure the Command Line Client to use MKS, execute the following command:

```
ccollab set 215 scm 205 mks
```

MKS Source-specific Options

Option	Description
--mks-host <value>	MKS Source server name
--mks-port <value>	MKS Source server port
--mks-user <value>	MKS Source user name

© 2003-2011 SmartBear Software

Option	Description
--mks-passwd <value>	MKS Source user password
--mks-expand-keywords	Whether to expand keywords in MKS source files

5.11.2.1 addchanges (for MKS Source)

```
Usage: ccollab [global-options] addchanges [--diffbranch <value> --upload-
comment <value>] <review> <file-spec> [<file-spec> ...]
```

`addchanges` - Attaches locally-modified files to a review

Command Options

Option	Required?	Description
--diffbranch <value>	N	(ClearCase only) specify either branch-name or "predecessor" to determine the earlier version for upload
--upload-comment <value>	N	Comment used to upload files (default is "Local changes")
<review>	Y	Must be either an integer review-id, 'new', 'ask', or 'last'
<file-spec> [<file-spec> ...]	Y	Local files or directories to scan for modified files

Uploads locally modified files controlled by MKS Source to the PeerReview Complete server.

The first argument is the review specifier, then any number of files or directories can be given. Files are selected as-is; directories are scanned recursively for locally-modified files, which might include edits, additions, deletions, branches, integrations, moves, copies, and so on.

After the scan is complete, a file list is presented in a graphical editor so you can review the files about to be uploaded and make any corrections you wish.

© 2003-2011 SmartBear Software

For example, to create a new review and add all changes in the current directory and below, plus the file foo.txt, you would use:

```
ccollab addchanges new . foo.txt
```

To upload modified files from the current working directory and all subdirectories to review 123:

```
ccollab addchanges 123 .
```

To upload file foo.txt and modified files from c:\dev\project into a brand new review:

```
ccollab addchanges new foo.txt c:\dev\project
```

5.11.2.2 addchangelist (for MKS Source)

```
Usage: ccollab [global-options] addchangelist <review> <changelist>
[<changelist> ...]
```

`addchangelist` - Attaches an atomic changelist to a review

Command Options

Option	Required?	Description
<review>	Y	Must be either an integer review-id, 'new', 'ask', or 'last'
<changelist> [<changelist> ...]	Y	SCM-specific ID of an atomic set of changes

Uploads Pending Change Packages or Change Packages from MKS Source to the PeerReview Complete server.

The first argument is the review specifier, subsequent arguments are the IDs of the Pending Change Packages or Change Packages to upload.

For example, to upload Pending Change Packages 1:2 and 3:2 to a new review:

© 2003-2011 SmartBear Software

ccollab addchangelist new 1:2 3:2

For example, to upload Change Packages 1:1 and 3:1 to review 111:

ccollab addchangelist 111 1:1 3:1

5.11.2.3 addversions (for MKS Source)

```
Usage: ccollab [global-options] addversions [--upload-comment <value>]
<review> [<version-spec> [<version-spec> ...]]
```

`addversions` - Attaches any 2 given versions to a review

Command Options

Option	Required?	Description
--upload-comment <value>	N	Comment used to upload files (default is "Local changes")
<review>	Y	Must be either an integer review-id, 'new', 'ask', or 'last'
<version-spec> [<version-spec> ...]	N	Versions to be reviewed are given by <path> <version> [<previous version>], where <path> is the filename or server path of the file, <version> is the version to be reviewed, and <previous version> is an optional version that <version> should be diff'ed against. If <previous version> is not specified, the predecessor version of <version> will be used. If no <version-spec> is given on the command line, then one or more version specs will be read from stdin, one <version-spec> per line as described above. Note that arguments containing spaces must be quoted to be parsed correctly

© 2003-2011 SmartBear Software

Uploads specific versions of a file or files to a review. A version-spec is given by a filename followed by a version number, and optionally, a previous version number:

```
file-name version-number [previous-version-number]
```

If no previous version number is explicity given, diffs will be generated with the predecessor version as determined from the SCM system.

Adding versions for a single file can be done with command line arguments given after the review number. To upload versions for multiple files, the file and versions may be submitted on lines read from standard input. If the path and versions are not given on the command line, the client will expect to read them from stdin. Each line read should be in the format shown above. Some example command lines for reading versions from stdin:

```
ccollab addversions 86753
ccollab addversions last < versionlist.txt
cat versionlist.txt | ccollab addversions new
```

When supplying the version list in the command line client editor or an input file, the keyword 'local' can be used to denote the version corresponding to the local version of the file. The 'local' keyword can only be used for the first version argument, not the previous version.

An example of specifying MKS revisions on the command line:

```
ccollab addversions review hello.c 1.95 1.88
```

© 2003-2011 SmartBear Software

5.12 Perforce Integration

This section describes PeerReview Complete integration with Perforce:

Perforce Server Integration 367

The PeerReview Complete server can pull submitted changelists directly from your Perforce server for review, without users needing to install any client programs. It can also enforce file content access permissions 368 (protections) configured in Perforce.

GUI Client 369

The GUI Client can upload Changelists into PeerReview Complete. You can upload Pending 370 or Submitted 370 Changelists, but you cannot upload the Default Changelist. The GUI Client can also upload arbitrary Perforce diffs 371, files in a Branch 372, or the difference between two Labels 372 or dates 373.

Command Line Client 374

The Command Line Client can upload Changelists into Command Line Client. You can upload Changelists Pending or Submitted Changelists, but you cannot upload the Default Changelist. The Command Line Client can also upload arbitrary Perforce diffs.

P4V and P4Win

PeerReview Complete comes with plug-ins to P4V and P4Win 381 that let you right-click on any changelist (pending or submitted) and add to a new or existing review.

Perforce Server Triggers

Perforce server-side triggers can ensure code is reviewed 383. There is also a trigger to automatically update a changelist 383 with information about the review of that changelist.

Perforce Changelist Renumbering

Perforce nearly always renumbers changelists upon submission, but when you're doing pre-commit reviews PeerReview Complete always has the pre-submit number. We've included a script you can customize that will ask Perforce what a changelist's number was before it was submitted, and update changelists in PeerReview Complete to have the new number. This requires a PeerReview Complete client and server **5.0.5005** or better, and a Perforce client and server **2007.3** or later. This script is intended to be run periodically via `cron` or a similar task scheduler. Perforce Changelist Renumbering Script

Supported Versions

Our integration uses your own Perforce command-line client (`p4`) to communicate with the server. We support all client and server versions later than **2002.1**.

Because we use client applications already present on your computer, we support all protocols, authentications, proxies, and other client configuration options you are currently using. This includes configuration from environment variables, `$P4CONFIG` files, `p4 set` registry values, and so forth.

Support for Branch / Integrate

PeerReview Complete fully supports Perforce's file branching and integration semantics.

If files in a changelist are marked for branching, they are not considered "changed." The file content itself is not changed; only the file paths are being changed.

© 2003-2011 SmartBear Software

If files in a changelist are being integrated, this works just like a regular change. Many customers choose to review integrations especially carefully since the changes might interact in unexpected ways.

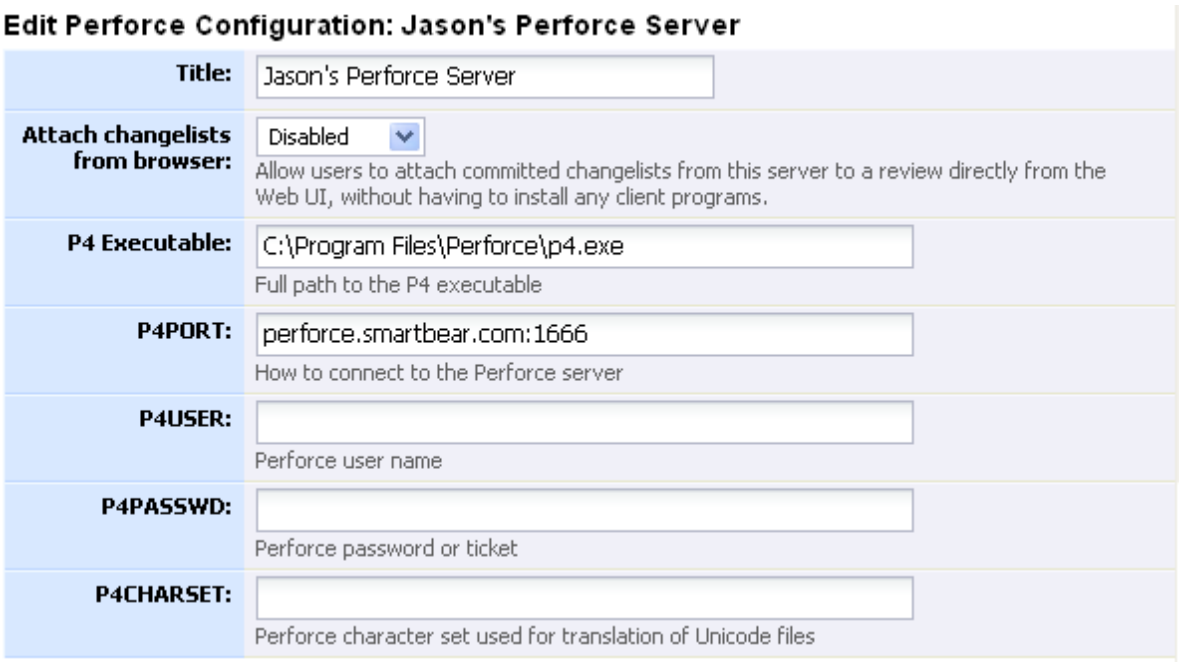
Support for Copy / Move
PeerReview Complete fully supports Perforce's file copy/move semantics. Thus, if a file is copied or moved rather than added from scratch, it will show up that way in the various user interfaces.

Support for Directory-level New/Delete/Copy/Move
PeerReview Complete partially supports Perforce's concept of directories (not just files) being altered.

All files underneath the directories in question will be scanned, uploaded and represented properly in the GUI. The directories themselves will*not* be shown in any GUI.

5.12.1 Perforce Server Integration

The PeerReview Complete server can be configured to communicate directly with your Perforce server. This allows users to review submitted changelists completely from the browser, without having to install any client programs. To enable this feature, create an entry for your Perforce server in the Version Control 101 tab of the administration interface. Version control server entries are also created automatically if one of the client programs uploads files from a server that doesn't match any of the currently configured servers.


Edit Perforce Configuration: Jason's Perforce Server
Title:
Jason's Perforce Server
Attach changelists from browser:
Disabled
Allow users to attach committed changelists from this server to a review directly from the Web UI, without having to install any client programs.
P4 Executable:
C:\Program Files\Perforce\p4.exe
Full path to the P4 executable
P4PORT:
perforce.smartbear.com:1666
How to connect to the Perforce server
P4USER:
Perforce user name
P4PASSWD:
Perforce password or ticket
P4CHARSET:
Perforce character set used for translation of Unicode files

Title The title is displayed to users, so it should be something that everyone will understand, even if they're going through proxies, VPNs, or other such things. When a version control server entry is created automatically, this is filled in with the server address (p4port) of the server.

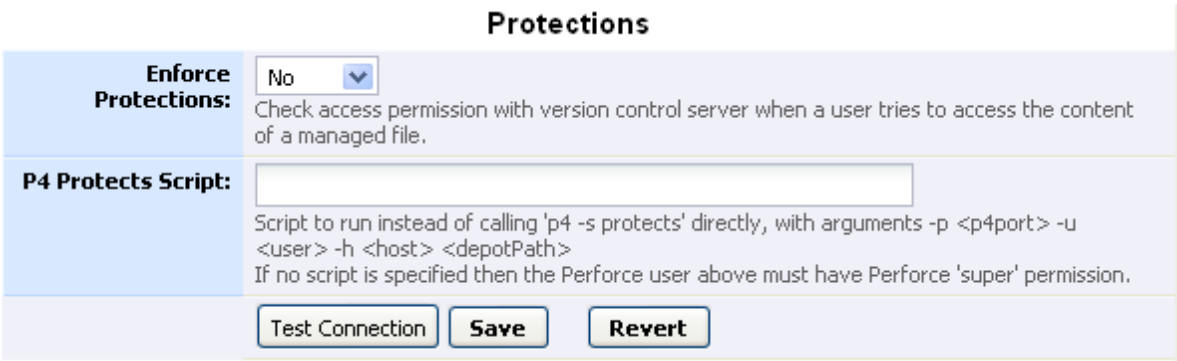
Attach changelists from browser If enabled, this feature lets users select submitted changelists to review directly from the web browser, without having to install any client programs.

© 2003-2011 SmartBear Software

P4 Executable	Full path to the P4 executable
P4PORT	Address to use to connect to the server. When a version control server entry i created automatically, this is filled in with the server address obtained from the client.
P4USER	Perforce user name
P4PASSWD	Perforce password or ticket
P4CHARSET	Perforce character set used for translation of Unicode files

Perforce Protections

The PeerReview Complete server can be configured to check Perforce protections with the Perforce server whenever a user tries to access a file.


Protections
Enforce Protections:
No
Check access permission with version control server when a user tries to access the content of a managed file.
P4 Protects Script:
Script to run instead of calling 'p4 -s protects' directly, with arguments -p <p4port> -u <user> -h <host> <depotPath>
If no script is specified then the Perforce user above must have Perforce 'super' permission.
Test Connection
Save
Revert

When enabled, PeerReview Complete will check access permission with your Perforce server whenever a user tries to access the content of a file managed by this server. Note that this assumes that the user's Perforce username is the same as their PeerReview Complete login.

To check Perforce protections the configured P4USER 368 must have Perforce "super" permission. If you don't want to provide an account with that level of permission, you can instead configure a script for PeerReview Complete to run instead of calling `p4 -s protects'` directly.

The script will be passed arguments `"-p <p4port> -u <user> -h <host> <depotPath>"`. For example, this script could be implemented as a Windows batch file:

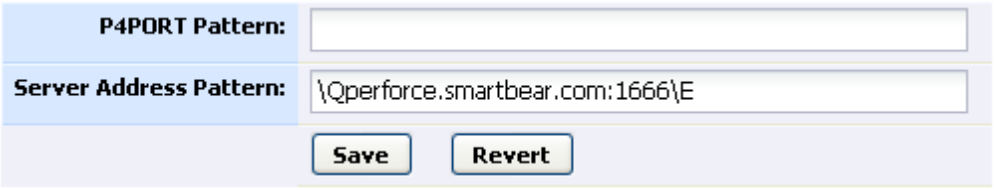
```
@p4 -u admin -s -p %2 protects -u %4 -h %6 %7
```

> The script must produce *exactly* the same stdin, stdout, and process error code as if PeerReview Complete ran `"p4 -s protects"` directly. Also make sure the script runs as fast as possible, because it will be invoked every time any user tries to view the contents a file from this server.

Client Configuration Mapping

You can supply java-style regular expressions to map changelists uploaded from our client tools to this Perforce server. It is important to set up these regular expressions so that files uploaded by the various PeerReview Complete client tools are correctly associated with this server-side Perforce configuration.

© 2003-2011 SmartBear Software



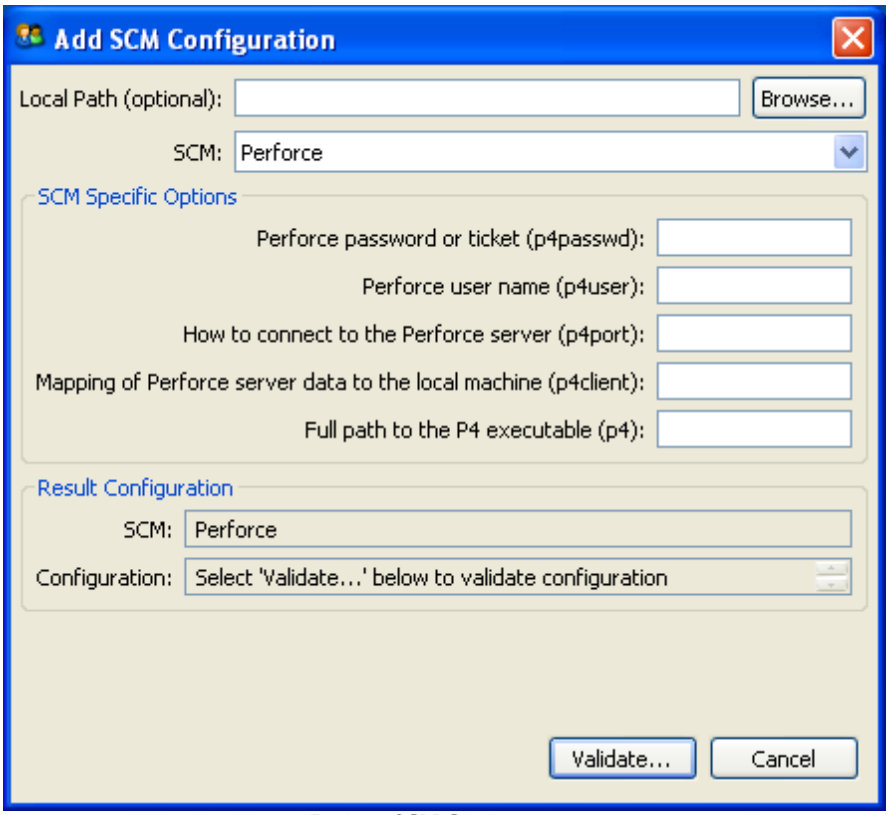
P4PORT Pattern Match on the client's configured P4PORT. Using the Server Address Pattern 369 is better, but clients before 5.0.5009 didn't upload "Server Address", so you can configure this as a fallback.

Server Address Pattern Match on the "Server Address" returned from running "`p4 -p <p4port> info`" on the client. This is generally more reliable than matching on the P4PORT, and works through Perforce proxy servers and other complications. When a version control server entry is created automatically, this is filled in with the server address obtained from the client.

5.12.2 GUI Client

Perforce-specific Options

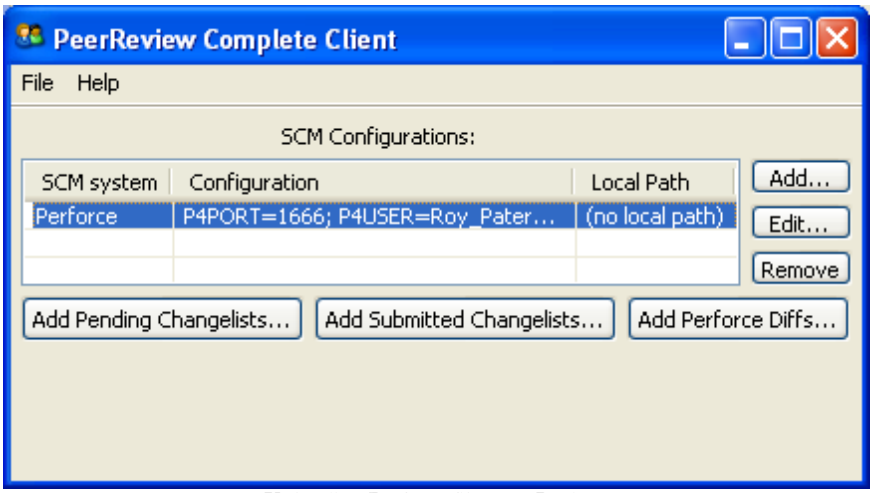
The SCM Configuration dialog 191 has several Perforce-specific options. These can be set as necessary to override Perforce options derived from the environment.



Perforce SCM Configuration

Uploading files to a Review

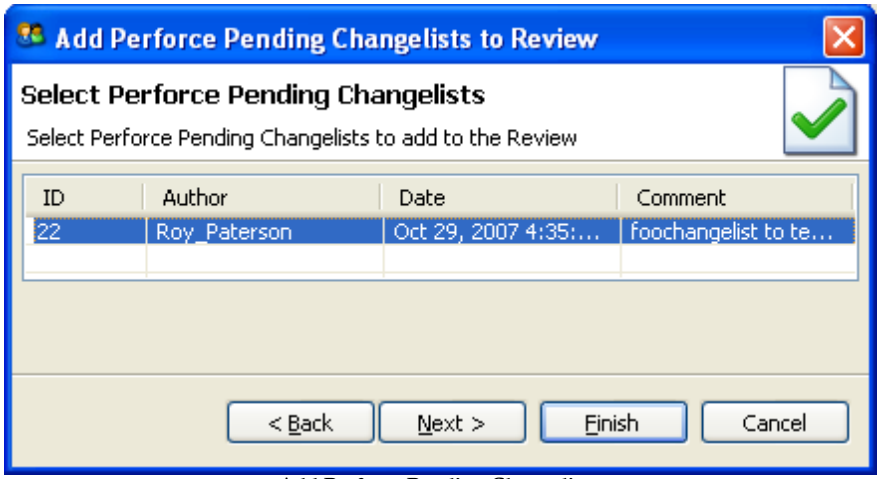
Selecting a Perforce SCM Configuration in the GUI Clientmain screen 190 causes several Add to Review buttons to appear. The *Add Pending Changelists...* 370 button uploads numbered, pending changelists. The *Add Submitted Changelists...* 370 button uploads submitted changelists. The *Add Perforce Diffs...* 371 button uploads arbitrary diffs, compares files in a Branch 372, or compares the difference between two Labels 372 or dates 373.



Uploading Perforce files to a Review

Add Perforce Pending Changelists

Press the *Add Perforce Pending Changelists...* button to upload the files in a Perforce pending changelist to the PeerReview Complete Server for review. You cannot upload the default changelist.

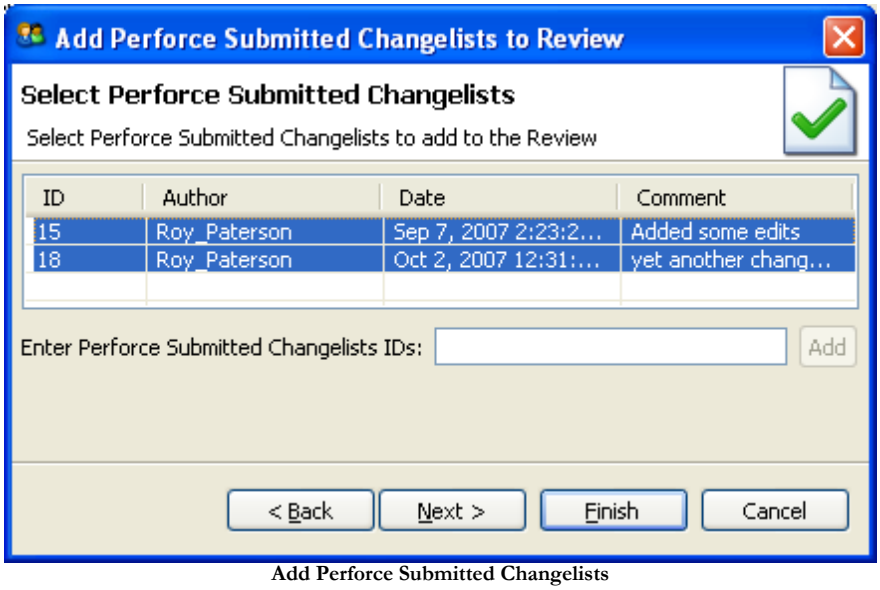


Add Perforce Pending Changelists

Add Perforce Submitted Changelists

Press the *Add Perforce Submitted Changelists...* button to upload the files in a Perforce submitted changelist to the PeerReview Complete Server for review.

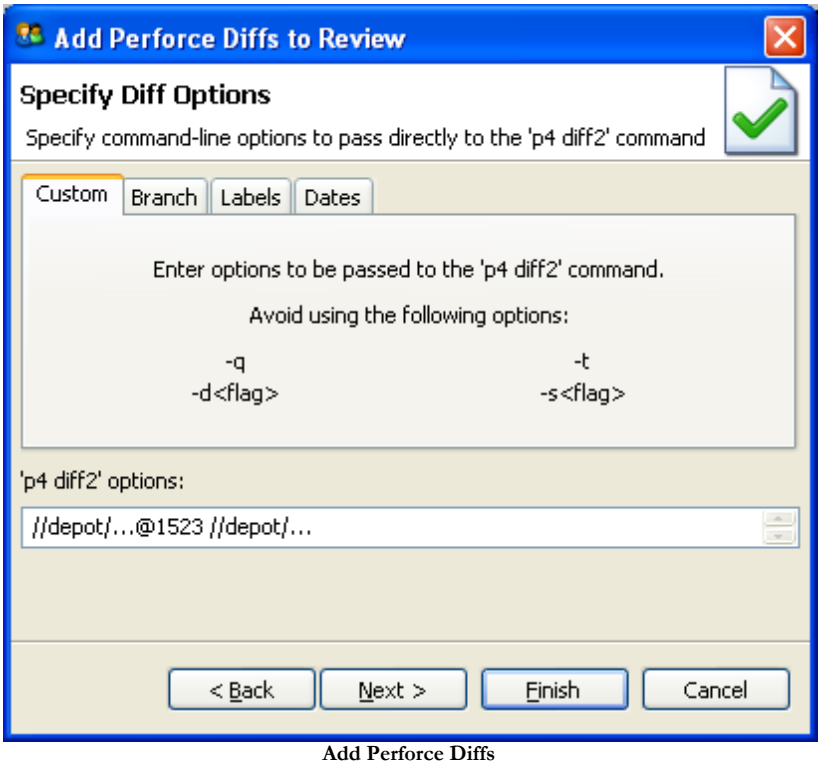
© 2003-2011 SmartBear Software



Add Perforce Submitted Changelists

Add Perforce Diffs

Press the *Add Perforce Diffs...* button to upload arbitrary Perforce diffs to the PeerReview Complete Server for review.



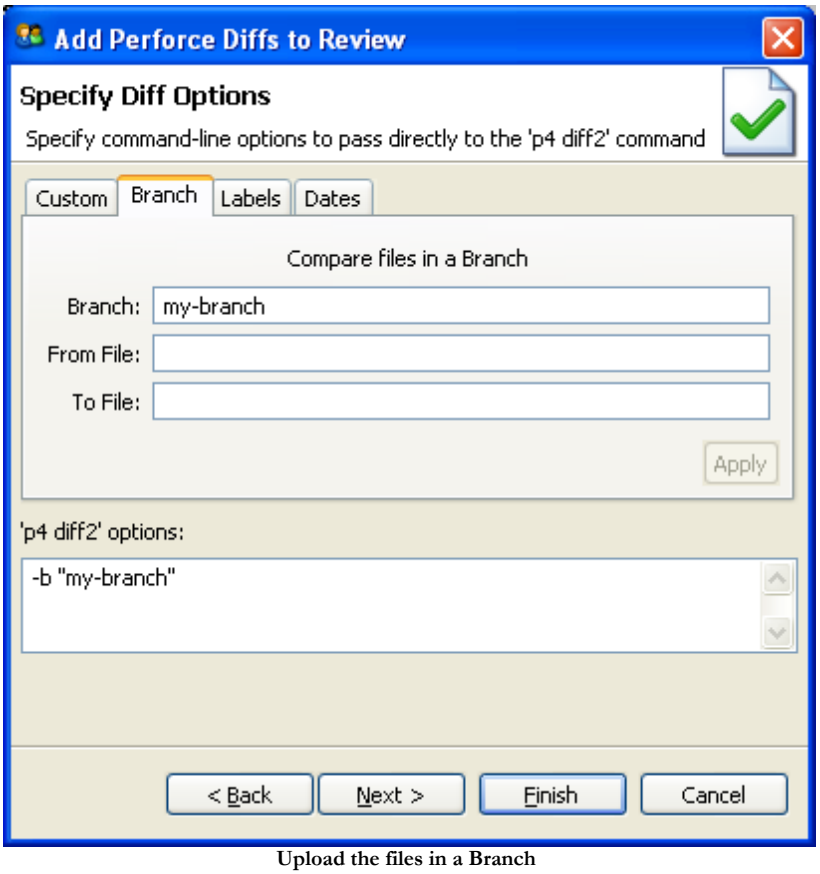
Add Perforce Diffs

You can enter arbitrary Perforce diff options, compare files in a Branch [372], or compare the difference between two Labels [372] or dates [373].

© 2003-2011 SmartBear Software

5.12.2.1 Comparing files in a Branch

Press the *Add Perforce Diffs...* 371 button on the main screen 190 and then select the *Branch* tab to upload the files in a Branch.

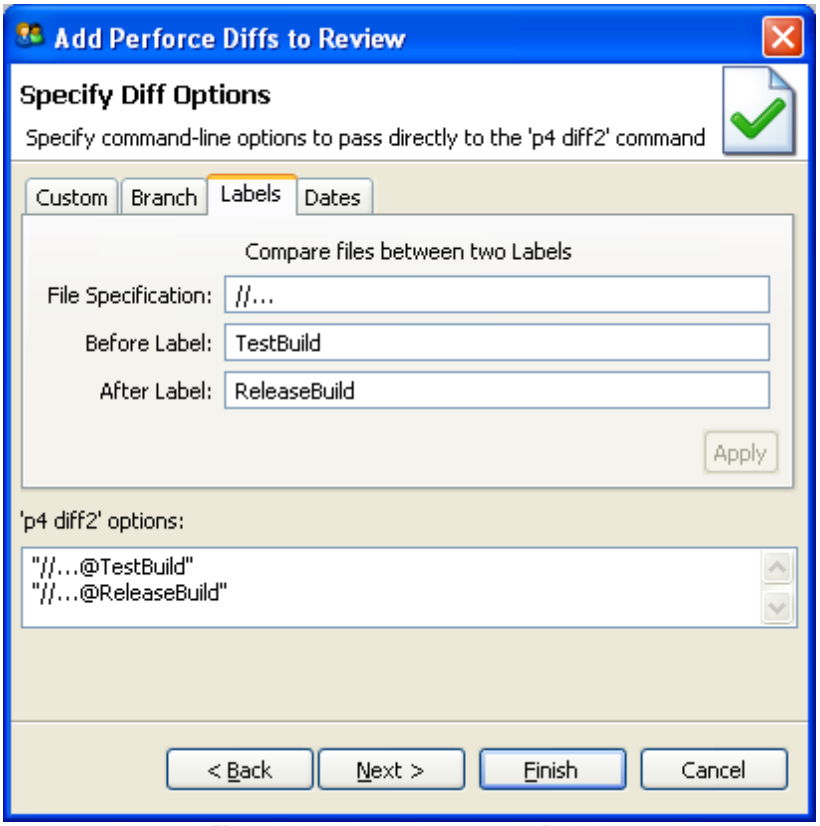


Upload the files in a Branch

5.12.2.2 Comparing two Labels

Press the *Add Perforce Diffs...* 371 button on the main screen 190 and then select the *Labels* tab to upload the difference between two Labels.

© 2003-2011 SmartBear Software

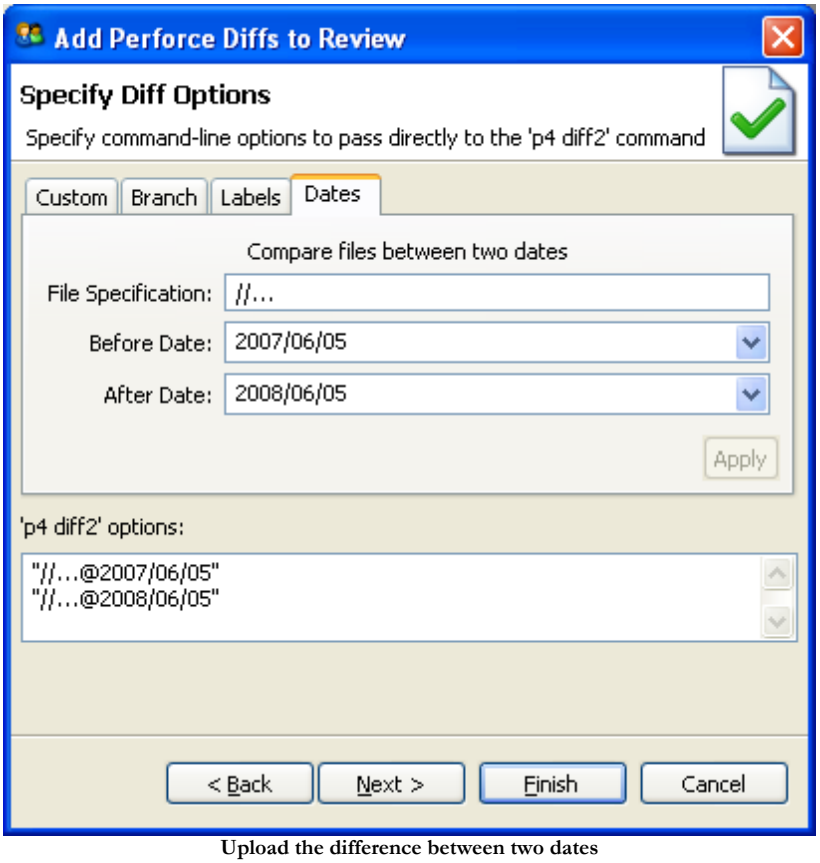


Upload the difference between two Labels

5.12.2.3 Comparing two dates

Press the *Add Perforce Diffs...* 371 button on the main screen 190 and then select the *Dates* tab to upload the difference between two dates.

© 2003-2011 SmartBear Software



Upload the difference between two dates

5.12.3 Command Line Client

Commands recommended for Perforce

ccollab addchangelist 375 - Attaches an atomic changelist to a review

ccollab addp4diffs 224 - Uploads diffs generated from p4 diff2 command

ccollab addversions 377 - Attaches any 2 given versions to a review

ccollab commit 379 - Commit changes in the review

ccollab addp4job 233 - Adds all numbered local changes that fix a job to the review

ccollab admin syncusers 235 - Synchronizes the Perforce user list with the PeerReview Complete server

ccollab admin changelist update-id 278 - Updates the changelist ID so PeerReview Complete reflects the renumbered Perforce changelist ID

The addchangelist 375 command will upload Perforce changelists into PeerReview Complete. You can upload changelists either before or after they are submitted, but you cannot upload the default changelist.

© 2003-2011 SmartBear Software

Configuration
In most cases, the Command Line Client can automatically detect your Perforce configuration. Try testing your configuration [197] to verify the configuration is detected correctly.

If the Command Line Client is unable to detect your Perforce configuration or you want to override the detected settings, you can manually specify Perforce settings using global options [207].

To manually configure the Command Line Client to use Perforce, execute the following command:

```
ccollab set [215] scm [205] perforce
```

Perforce-specific Options

Option	Description
--p4 <value>	Full path to the P4 executable
--p4port <value>	How to connect to the Perforce server
--p4user <value>	Perforce user name
--p4passwd <value>	Perforce password or ticket
--p4client <value>	Mapping of Perforce server data to the local machine
--p4-ignore-integration-history <value>	Ignore integration history when calculating predecessor
--p4charset <value>	Perforce character set used for translation of Unicode files
--p4-require-empty-default-changelist	If true, don't allow uploads if the default changelist contains files
--p4-specify-command-charset <value>	Should a character set be specified for communication with Perforce

If your Perforce server requires ticket-based authentication (server security level 3) then the configured p4passwd will be used to automatically issue 'p4 login' and acquire a new ticket as necessary.

If you want to ignore the integration history of files when determining the previous version of the file and look only at the path you should set p4-ignore-integration-history to yes like so:

```
ccollab set p4-ignore-integration-history yes
```

5.12.3.1 addchangelist (for Perforce)

```
Usage: ccollab [global-options] addchangelist <review> <changelist>
[<changelist> ...]
```

© 2003-2011 SmartBear Software

`addchangelist` - Attaches an atomic changelist to a review

Command Options

Option	Required?	Description
<review>	Y	Must be either an integer review-id, 'new', 'ask', or 'last'
<changelist> [<changelist> ...]	Y	SCM-specific ID of an atomic set of changes

Uploads Pending Changelists or Submitted Changelists from Perforce to the PeerReview Complete server.

The first argument is the review specifier, subsequent arguments are the IDs of the Pending Changelists or Submitted Changelists to upload.

You cannot specify the default changelist.

For example, to upload Pending Changelists @4321 and @7568 to a new review:

```
ccollab addchangelist new 4321 7568
```

For example, to upload Submitted Changelists @5432 and @12654 to review 111:

```
ccollab addchangelist 111 5432 12654
```

5.12.3.2 addp4diffs

```
Usage: ccollab [global-options] addp4diffs [--upload-comment <value>]
<review> [<p4-diff-arg> [<p4-diff-arg> ...]]
```

`addp4diffs` - Uploads diffs generated from p4 diff2 command

Command Options

© 2003-2011 SmartBear Software

Option	Required?	Description
--upload-comment <value>	N	Comment used to upload files (defaults to command-line arguments)
<review>	Y	Must be either an integer review-id, 'new', 'ask', or 'last'
<p4-diff-arg> [<p4-diff-arg> ...]	N	Options which should be passed to the p4 diff2 command

Upload differences between arbitrary versions of files in Perforce using 'p4 diff2'.

Do not use diff arguments that affect the diff output such as -q, -t, -d<flag>, or -s<flag>. The PeerReview Complete command-line client will automatically select an output format that ensures you will get all the data you need on the server.

Examples:

ccollab addp4diffs 698 //depot/file1 //depot/file2

ccollab addp4diffs new //depot/...@1523 //depot/...

ccollab addp4diffs 698 -b my-branch

5.12.3.3 addversions (for Perforce)

```
Usage: ccollab [global-options] addversions [--upload-comment <value>]
<review> [<version-spec> [<version-spec> ...]]
```

`addversions` - Attaches any 2 given versions to a review

Command Options

Option	Required?	Description
--upload-comment <value>	N	Comment used to upload files (default is "Local changes")
<review>	Y	Must be either an integer review-id, 'new', 'ask', or 'last'

© 2003-2011 SmartBear Software

Option	Required?	Description
<version-spec> [<version-spec> ...]	N	Versions to be reviewed are given by <path> <version> [<previous version>], where <path> is the filename or server path of the file, <version> is the version to be reviewed, and <previous version> is an optional version that <version> should be diff'ed against. If <previous version> is not specified, the predecessor version of <version> will be used. If no <version-spec> is given on the command line, then one or more version specs will be read from stdin, one <version-spec> per line as described above. Note that arguments containing spaces must be quoted to be parsed correctly

Uploads specific versions of a file or files to a review. A version-spec is given by a filename followed by a version number, and optionally, a previous version number:

file-name version-number [previous-version-number]

If no previous version number is explicity given, diffs will be generated with the predecessor version as determined from the SCM system.

Adding versions for a single file can be done with command line arguments given after the review number. To upload versions for multiple files, the file and versions may be submitted on lines read from standard input. If the path and versions are not given on the command line, the client will expect to read them from stdin. Each line read should be in the format shown above. Some example command lines for reading versions from stdin:

© 2003-2011 SmartBear Software

ccollab addversions 86753

ccollab addversions last < versionlist.txt

cat versionlist.txt | ccollab addversions new

When supplying the version list in the command line client editor or an input file, the keyword 'local' can be used to denote the version corresponding to the local version of the file. The 'local' keyword can only be used for the first version argument, not the previous version.

5.12.3.4 commit (for Perforce)

```
Usage: ccollab [global-options] commit [--comment <value> --dismiss-only --
force] <review>
```

`commit` - Commit changes in the review

Command Options

Option	Required?	Description
--comment <value>	N	Comment for reviewed changes
--dismiss-only	N	Just dismiss the Action Item
--force	N	Ignore potential problems
<review>	Y	Must be either an integer review-id, 'ask', or 'last'

Commit the changes from a pre-commit review to sourcecontrol. Be sure to include a relevant comment.

5.12.3.5 addp4job

```
Usage: ccollab [global-options] addp4job <review> <job-name>
```

`addp4job` - Adds all numbered local changes that fix a job to the review

© 2003-2011 SmartBear Software

Command Options

Option	Required?	Description
<review>	Y	Must be either an integer review-id, 'new', 'ask', or 'last'
<job-name>	Y	Perforce changes that fix this job will be added to the review

5.12.3.6 syncusers

```
Usage: ccollab [global-options] admin syncusers
```

`syncusers` - Synchronizes the Perforce user list with the PeerReview Complete server

Causes all users from Perforce to be mirrored into PeerReview Complete.

The algorithm is smart enough not to overwrite existing users, only adding new users, so this can be run periodically by a script to keep the lists in sync.

This is not necessary -- or desirable -- if you are using LDAP authentication.

You must be logged in as an administrator to execute this command.

5.12.3.7 update-id

```
Usage: ccollab [global-options] admin changelist update-id <old-id> <new-
id>
```

`update-id` - Updates the changelist ID so PeerReview Complete reflects the renumbered Perforce changelist ID

Command Options

© 2003-2011 SmartBear Software

Option	Required?	Description
<old-id>	Y	
<new-id>	Y	

Updates the changelist ID using the provided configuration to match an existing Version Control configuration on the PeerReview Complete server to update all references to the old changelist ID. For example:

ccollab --scm p4 --p4port p4server:1666 admin changelist update-id 123 147

5.12.4 P4V / P4Win Integration

The PeerReview Complete client installer includes an integration point with **P4V** and **P4Win**. When you right-click on one or more Changelists in the GUI a new menu item appears at the bottom of the Tools menu allowing you to add those Changelists to a Review.

Installation

If you installed the client yourself and opted to "Configure Addons To Perforce Visual Tools 185", then you should not have to manually configure the Perforce visual tools integrations. However, if the PeerReview Complete client was installed by an administrator on your computer, or the configuration failed at install time, you can manually configure them as follows:

P4V Installation

Go to the "`Tools`" menu in P4V and choose "`Manage Custom Tools...`". In the dialog box click "`Import Tools...`". In the ensuing open-file dialog, navigate to the PeerReview Complete Client installation directory and select the `P4V-Tools-Import.txt` or `P4V-Tools-Import.xml` file.

P4Win Installation

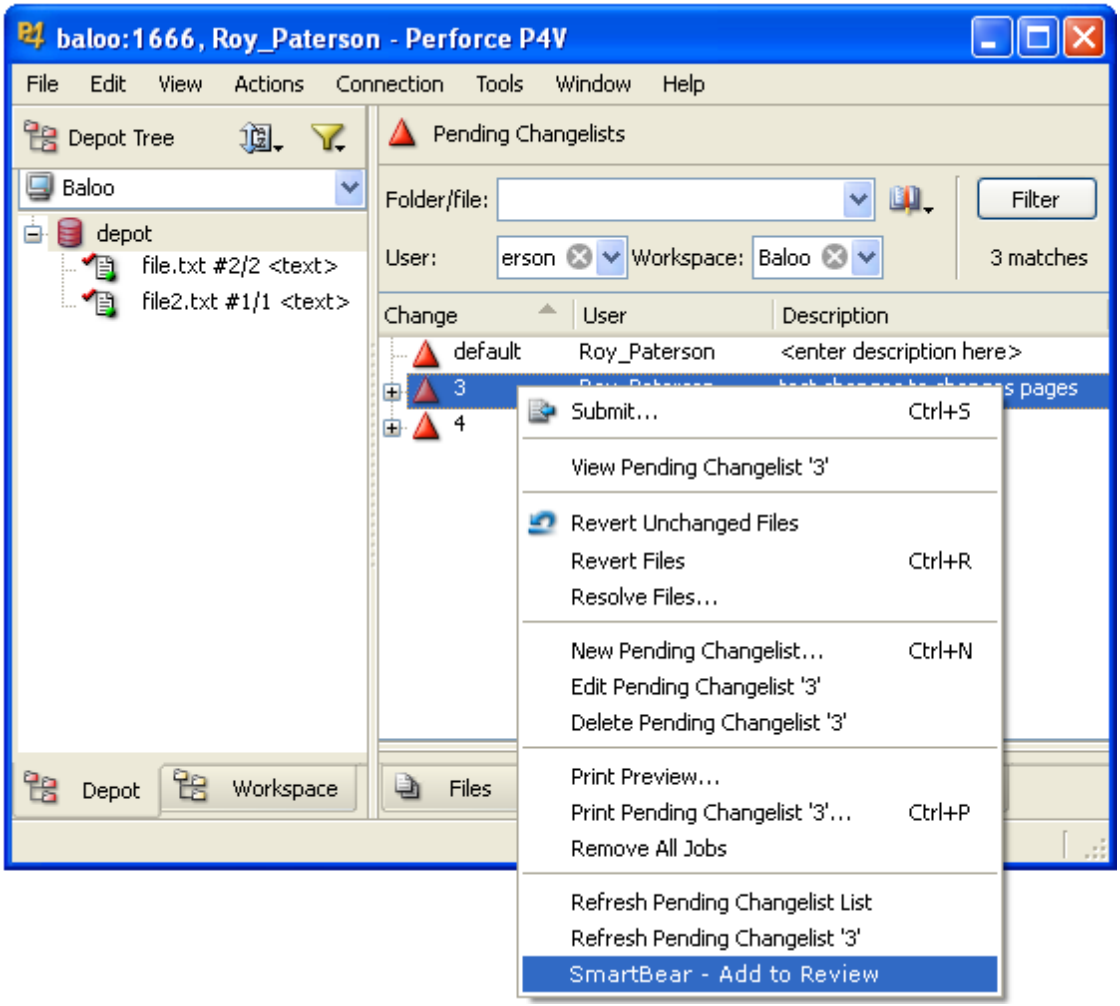
Go to the "`Tools`" menu in P4Win and choose "`Import...`". In the ensuing open-file dialog, navigate to the PeerReview Complete Client installation directory and select the `P4Win-Tools-Import.txt` file.

There should now be a menu item under "`Tools`" with the text "`PeerReview Complete - Add to Review`". It will be dimmed unless a Changelist is selected.

© 2003-2011 SmartBear Software

Warning: The plug-ins will work only if the Perforce command-line tool (p4) is in your PATH environment variable.

It is also available in the pop-up menu when you right-click a Changelist:



Working with multiple Perforce workspaces and servers

The P4V/P4Win plug-ins work perfectly and automatically with multiple Perforce servers and servers.

All the Perforce connectivity parameters are taken from the UI, so whatever workspace, server, and authentication you're currently using when you right-click the item will be used automatically be the plug-in.

None of this affects the behavior or configuration of the command-line client 197, so you can freely use the plug-ins without disturbing that configuration.

© 2003-2011 SmartBear Software

Troubleshooting
If you are experiencing problems with the P4V/P4Win plug-ins, it will help SmartBear technical support if you send in a debugging log. To do this, edit the custom tool configuration (Tools > Manage Custom Tools > Edit), and add a --debug argument to the beginning of the arguments list. This will create a ccollab.log file in the "current working directory" which is either the Perforce install directory or the "Start in" directory specified in the Custom Tool configuration.

Please send the log file along with a full description of what you were trying to do when the error occurred to support@smartbear.com.

5.12.5 Perforce Server Triggers

Triggers recommended for Perforce

ccollab admin trigger ensure-review-started 383 - Changelist cannot be submitted until review of this changelist exists

ccollab admin trigger ensure-reviewed 385 - Changelist cannot be submitted until review of this changelist has been completed

ccollab admin trigger ensure-content-reviewed 266 - Changelist cannot be submitted until review completed & file list/content is unchanged

ccollab admin trigger update-changelist 268 - Automatically update Perforce changelist with information about the review of that changelist

The ensure-review-started 383, ensure-reviewed 385, or ensure-content-reviewed 266 triggers ensure that changelists cannot be submitted unless certain conditions are met. If a user attempts to submit a changelist that doesn't meet those conditions, an error message describing the unfulfilled conditions will be displayed and the changelist will not be submitted. If the conditions are met, the submit will be allowed to continue normally.

The update-changelist 268 trigger adds or updates a one-line description in the changelist description automatically every time a changelist is edited or submitted . This means users can see the data update whenever they edit the description text. This trigger uses the `${...}` variable substitution subsystem 119 to let you insert dynamic data, which could include the review id 119, participants 120, and so on. See the substitution system topic for details on what variables you can include.

For more information about Perforce triggers in general, see Chapter 6 of the Perforce System Administrator's Guide.

5.12.5.1 ensure-review-started (for Perforce)

```
Usage: ccollab [global-options] admin trigger ensure-review-started [--
ignore-integrate --review-id-regex <value>] <changelist-id>
```

`ensure-review-started` - Changelist cannot be submitted until review of this changelist exists

© 2003-2011 SmartBear Software

Command Options

Option	Required?	Description
--ignore-integrate	N	Allow integration changes to proceed without review
--review-id-regex <value>	N	Regular Expression that identifies Review ID in commit comment
<changelist-id>	Y	Changelist ID

This trigger ensures that a Review has been started before changes can be committed to version control.

If a Review has not been started, the commit will fail and the user will get an error message explaining that the changes need to be reviewed before commit.

To install this trigger edit your Perforce Triggers list (see Perforce manual for details) and add a line like the following (except supply your installation directory for the ccollab application, replace <values> as appropriate, and put everything on one line even though the following text is word-wrapping):

ccollab submit //depot/... "/usr/bin/ccollab --url <collabUrl> --user <collabUser> --password <collabPasswd> --scm perforce --p4port <p4port> --p4user <p4user> --p4passwd <p4passwd> admin trigger ensure-review-started %changelist%"

On Unix systems the Perforce trigger system incorrectly splits up arguments by whitespace ignoring single quotes and other standard ways of indicating that data with spaces is actually a single argument. The solution is to use a separate bash script to call our trigger application.

You must supply credentials for a Perforce user that has at least read-only access.

Changelists that result from creating a new branch are ignored. Creating a new branch does not involve a code change, so it does not require review.

Integrating changes from one branch into another branch does make code changes and can introduce bugs, so by default this type of change is handled by the trigger. However, because integration changes frequently involve a very large number of files and because often it is necessary to integrate changes to many branches, this would require cumbersome reviews to be performed many times.

© 2003-2011 SmartBear Software

For this reason, the '--ignore-integrate' option can be used to ignore changelists consisting of only integration changes. In this case, enforcement of review happens externally to the system: for example, a review could be performed on one branch, but other branches could be just directly integrated.

5.12.5.2 ensure-reviewed (for Perforce)

```
Usage: ccollab [global-options] admin trigger ensure-reviewed [--ignore-
integrate --review-id-regex <value>] <changelist-id>
```

`ensure-reviewed` - Changelist cannot be submitted until review of this changelist has been completed

Command Options

Option	Required?	Description
--ignore-integrate	N	Allow integration changes to proceed without review
--review-id-regex <value>	N	Regular Expression that identifies Review ID in commit comment
<changelist-id>	Y	Changelist ID

This trigger ensures that a Review has been completed before changes can be committed to version control.

If a Review has not been completed, the commit will fail and the user will get an error message explaining that the changes need to be reviewed before commit.

To install this trigger edit your Perforce Triggers list (see Perforce manual for details) and add a line like the following (except supply your installation directory for the ccollab application, replace <values> as appropriate, and put everything on one line even though the following text is word-wrapping):

ccollab submit //depot/... "/usr/bin/ccollab --url <collabUrl> --user <collabUser> --password <collabPasswd> --scm perforce --p4port <p4port> --p4user <p4user> --p4passwd <p4passwd> admin trigger ensure-reviewed %changelist%"

On Unix systems the Perforce trigger system incorrectly splits up arguments by whitespace ignoring single quotes and other standard ways of indicating that data with spaces is actually a single argument. The solution is to use a separate bash script to call our trigger application.

© 2003-2011 SmartBear Software

You must supply credentials for a Perforce user that has at least read-only access.

Changelists that result from creating a new branch are ignored. Creating a new branch does not involve a code change, so it does not require review.

Integrating changes from one branch into another branch does make code changes and can introduce bugs, so by default this type of change is handled by the trigger. However, because integration changes frequently involve a very large number of files and because often it is necessary to integrate changes to many branches, this would require cumbersome reviews to be performed many times.

For this reason, the '--ignore-integrate' option can be used to ignore changelists consisting of only integration changes. In this case, enforcement of review happens externally to the system: for example, a review could be performed on one branch, but other branches could be just directly integrated.

5.12.5.3 ensure-content-reviewed

```
Usage: ccollab [global-options] admin trigger ensure-content-reviewed [--
ignore-integrate --no-keywords --review-id-regex <value>] <changelist-id>
```

`ensure-content-reviewed` - Changelist cannot be submitted until review completed & file list/ content is unchanged

Command Options

Option	Required?	Description
--ignore-integrate	N	Allow integration changes to proceed without review
--no-keywords	N	Disallow keyword expansion
--review-id-regex <value>	N	Regular Expression that identifies Review ID in commit comment
<changelist-id>	Y	Changelist ID

This trigger ensures that a Review has been completed and the file list and file content is unchanged before changes can be submitted to version control.

© 2003-2011 SmartBear Software

If a Review has not been completed, or the Review contains different files or file contents, the submit will fail and the user will get an error message explaining that the changes need to be reviewed before submit.

For Perforce, to install this trigger edit your Perforce Triggers list (see Perforce manual for details) and add a line like the following (except supply your installation directory for the ccollab application, replace <values> as appropriate, and put everything on one line even though the following text is word-wrapping):

ccollab content //depot/... "/usr/bin/ccollab --url <collabUrl> --user <collabUser> --password <collabPasswd> --scm perforce --p4port <p4port> --p4user <p4user> --p4passwd <p4passwd> admin trigger ensure-content-reviewed %changelist%"

On Unix systems the Perforce trigger system incorrectly splits up arguments by whitespace ignoring single quotes and other standard ways of indicating that data with spaces is actually a single argument. The solution is to use a separate bash script to call our trigger application.

For Perforce, you must supply credentials for a Perforce user that has at least read-only access.

For Perforce, changelists that result from creating a new branch are ignored. Creating a new branch does not involve a code change, so it does not require review.

For Perforce, integrating changes from one branch into another branch does make code changes and can introduce bugs, so by default this type of change is handled by the trigger. However, because integration changes frequently involve a very large number of files and because often it is necessary to integrate changes to many branches, this would require cumbersome reviews to be performed many times.

For this reason, the '--ignore-integrate' option can be used to ignore changelists consisting of only integration changes. In this case, enforcement of review happens externally to the system: for example, a review could be performed on one branch, but other branches could be just directly integrated.

5.12.5.4 update-changelist

```
Usage: ccollab [global-options] admin trigger update-changelist [--
description-prefix <value> --description-template <value>] <changelist-
spec>
```

`update-changelist` - Automatically update Perforce changelist with information about the review of that changelist

© 2003-2011 SmartBear Software

Command Options

Option	Required?	Description
--description-prefix <value>	Y	Prefix text for review line in changelist description
--description-template <value>	Y	Template text for review line in changelist description
<changelist-spec>	Y	%changelist% for 'commit' trigger, or %formfile% for 'form-in change' trigger

If you are reviewing changelists prior to submit, you might want some text in your changelist description saying the changelist was reviewed, which review ID, participants, and so on.

Running this command on the 'submit' trigger causes a description of the review to be added or updated in the changelist text when a pending changelist is submitted. If you also run this trigger on 'form-in change' then users can see the data update whenever they edit the changelist.

To install this trigger edit your Perforce Triggers list (see Perforce manual for details) and add a line like the following (except supply your installation directory for the ccollab application, replace <values> as appropriate, and put everything on one line even though the following text is word-wrapping):

ccollabupdatechangelist change-submit //depot/... "/usr/bin/ccollab --url <collabUrl> --user <collabUser> --password <collabPasswd> --scm perforce --p4port <p4port> --p4user <p4user> --p4passwd <p4passwd> --p4client <p4client> admin trigger update-changelist --description-prefix 'Review: ' --description-template 'ID ${review.id} by ${review.participants.rolename}' %changelist%

Optionally you can also add on the next line:

ccollabupdatechangelist form-in change "/usr/bin/ccollab --url <collabUrl> --user <collabUser> --password <collabPasswd> --scm perforce --p4port <p4port> --p4user <p4user> --p4passwd <p4passwd> admin trigger update-changelist --description-prefix 'Review: ' --description-template 'ID ${review.id} by ${review.participants.rolename}' %formfile%

Note - This example works under Windows, but under Unix the Perforce trigger system parses this line incorrectly, splitting up arguments by whitespace even though they are quoted using single-quotes.

© 2003-2011 SmartBear Software

The solution is to use a separate shell script file to call our code and to call that script from the Perforce trigger.

The --description-prefix prefix option is the prefix text that means: This is a line that should be updated with review information. The trigger is allowed to replace everything after that prefix with review information. If a changelist doesn't have a line with this prefix, a new line is appended to the end of the changelist description; otherwise the information is updated in-place.

The --description-template option is a template that will be inserted after the prefix. The template gives you a chance to decide which information about the review you want to include, in what order, and with any type of textual separators.

The template uses the ${...} variable substitution subsystem to let you insert dynamic data. In the example given above, the review ID and a list of participants is inserted between some English text.

If there is a review in the system that matches this changelist, the text is always updated either in-place or appended as described. On the other hand, if there is not already a review, the description is updated only if the prefix is already present; if there's no current review and no prefix than nothing is appended to the description.

© 2003-2011 SmartBear Software

5.13 SourceGear Vault Integration

This section describes PeerReview Complete integration with SourceGear Vault.

Vault Integration

PeerReview Complete supports SourceGear Professional and Standard Editions, and the earlier incarnations of Vault and Fortress. The integration is provided through SourceGear's Vault client API, which is packaged for PeerReview Complete and available on our website for downloading. Choose from the following links for the to download the SourceGear client API for your Vault product:

SourceGear API for Vault 5.1 Professional Edition

SourceGear API for Vault 5.1 Standard Edition

SourceGear API for Vault 5.0 Standard Edition

SourceGear API for Fortress 2.0

SourceGear API for Vault 4.1

After downloading, the file must be renamed and copied under the installation directory as:

```
  ui\plugins\com.smartbear.collaborator_6.5.xxxx\libs_single_instance
\vault_client_api.jar
```

where `XXXX` is the build number of the release.

Reinstallation

Reinstalling the PeerReview Complete client will delete this directory, so you might want to temporarily move the `vault_client_api.jar` file to another directory before reinstallation.

Other Supported Versions

Generally, there are compatible releases of the SourceGear API for all Vault versions and we should be able to support your specific version. If there is no link above corresponding to your version of Vault, please contact us at support@smartbear.com and we will make it available.

Windows Installation Notes

Installing under Windows may require a temporary workaround for the SourceGear Vault client API. If your system uses a workgroup instead of a domain, the `USERDNSDOMAIN` environment will not be set to a value. This variable is required for the Vault client API to understand the correct username of the client. You can determine the value of `USERDNSDOMAIN` by typing `set` at a command prompt and looking through the environment variables. If the variable is not set, it *must* be defined before running the Command Line Client, or items in Vault's pending change set will not be found. The variable must be set to the name of the local workgroup, typically `mshome` or `workgroup`. You can find the workgroup name of the system on the General tab of the System Properties dialog.

There are two ways to set this environment variable for the Command Line Client. Create or edit the `ccollab.vmoptions` file in the Command Line Client installation directory, and include the line (with your workgroup name substituted):

```
userdnsdomain=workgroup-name
```

Note: the last line of this file *must* be followed by a line feed.

© 2003-2011 SmartBear Software

Another way to define this variable is in the User environment variable space. On the Advanced tab of System Properties, select the Environment Variables button and create a User variable named `USERDNSDOMAIN` with a value of the local workgroup name.

GUI Client 392

The GUI Client integrates with SourceGear Vault. It can find and upload Vault pending change set items, or already committed Vault source files.

Command Line Client 394

The Command Line Client integrates with SourceGear Vault. It can find and upload Vault pending change set items, or already committed Vault source files.

Troubleshooting

Enabling debugging in the PeerReview Complete clients when Vault is used will create a `ccollab_vault.log` file, in the current directory for theCommand Line Client or the installation directory for the GUI Client. This file contains trace logging from the SourceGear Vault API, which in addition to the PeerReview Complete client debugging log, can be useful for troubleshooting Vault issues. When submitting a client debugging log to technical support for a Vault issue, please also include this file.

Getting an "OutOfMemoryError: Java heap space" Error?

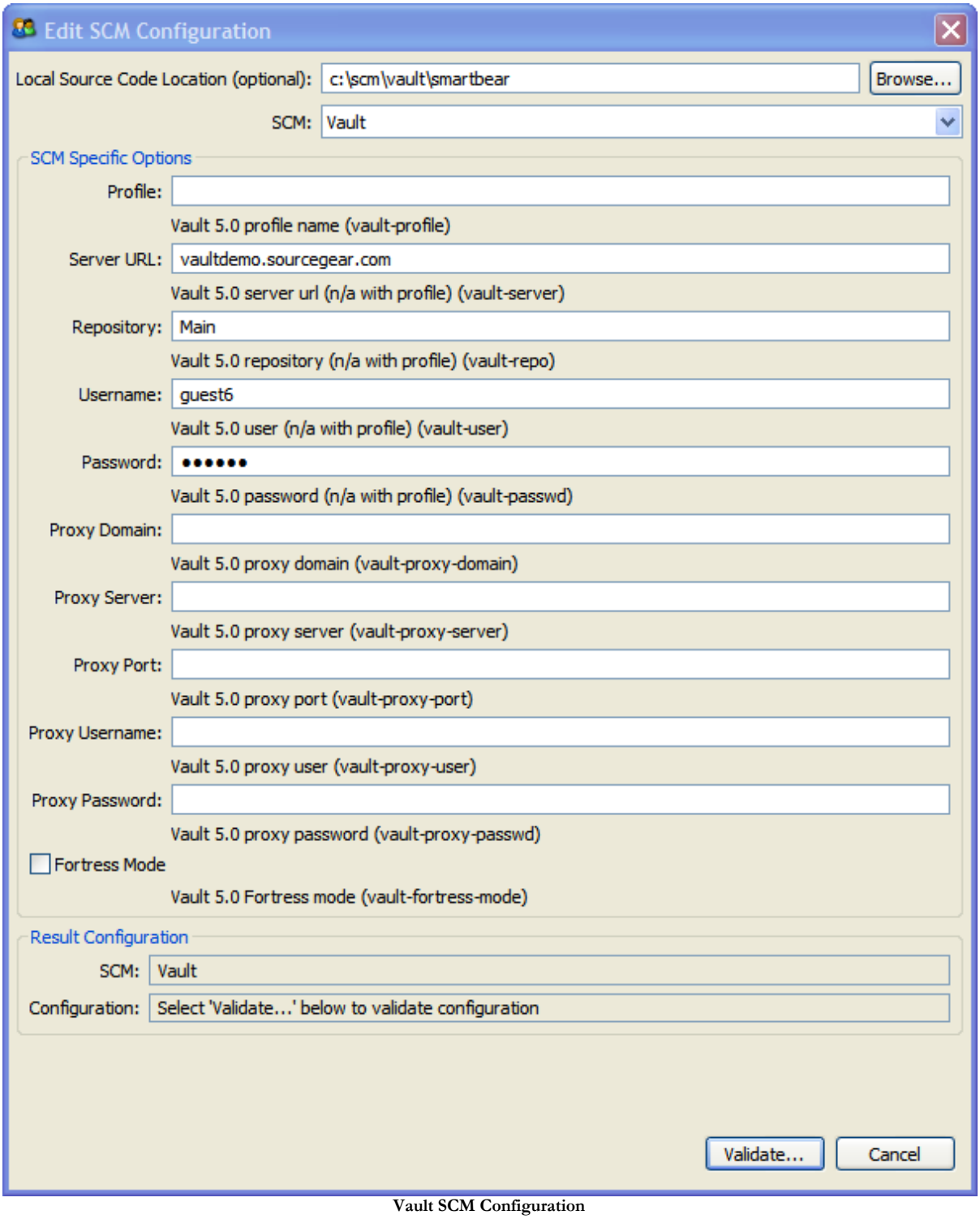
The SourceGear Vault API, like the SourceGear Vault client, utilizes local repository caching to reduce traffic to the server. When connecting to large Vault repositories with the PeerReview Complete clients, the memory requirements can exceed the default Java heap space available. Refer to the Java heap size 526 section of the manual for increasing the allocation. Increasing the available heap to 256MB of heap is a good starting point.

© 2003-2011 SmartBear Software

5.13.1 GUI Client

Vault-specific Options

The SCM Configuration dialog 191 has quite a few Vault-specific options.

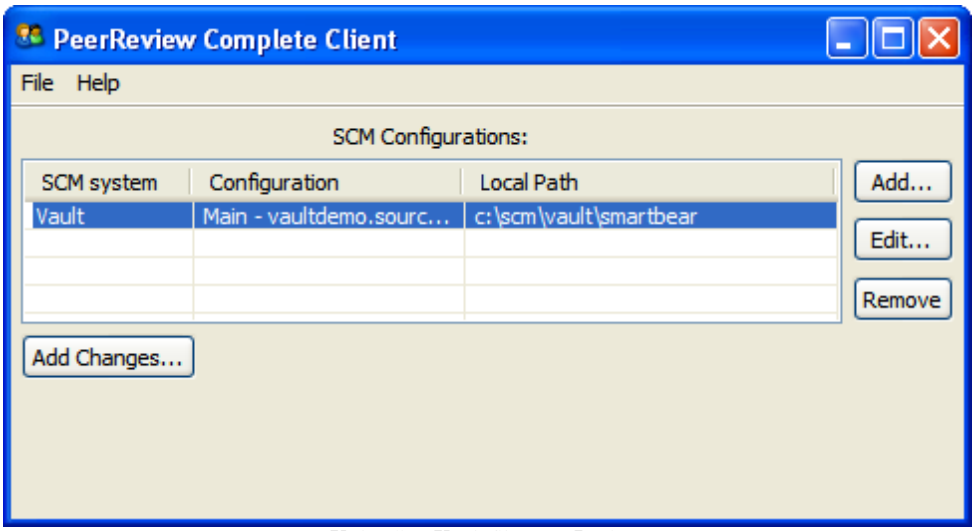


Vault SCM Configuration

© 2003-2011 SmartBear Software

Uploading files to a Review

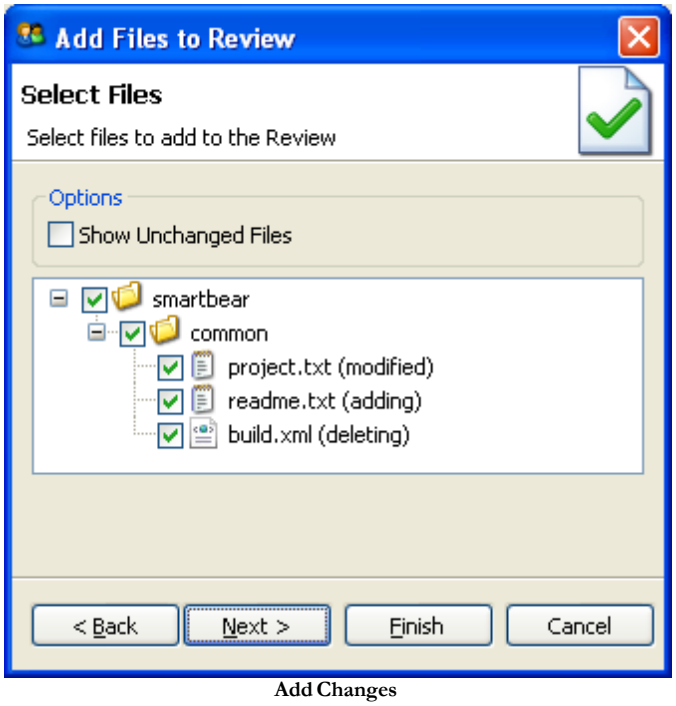
Selecting a Vault SCM Configuration in the GUI Client main screen 190 causes the *Add Changes...* button to appear. The *Add Changes...* 393 button uploads items in the Vault pending change set.



Uploading Vault files to a Review

Add Changes

Press the *Add Changes...* button to upload items in the Vault pending change set to the PeerReview Complete Server for review.



Add Changes

© 2003-2011 SmartBear Software

5.13.2 Command Line Client

Command recommended for Vault

ccollab addchanges 395 - Attaches locally-modified files to a review

The addchanges 308 command will find and upload Vault pending change set items.

Configuration

In most cases, the Command Line Client can automatically detect your Vault configuration. Try testing your configuration 197 to verify the configuration is detected correctly.

If the Command Line Client is unable to detect your Vault configuration or you want to override the detected settings, you can manually specify Vault settings using global options 204.

To manually configure the Command Line Client to use Vault, execute the following command:

```
ccollab set 215 scm 205 vault
```

Vault-specific Options

Option	Description
--vault-profile <value>	Vault profile name
--vault-server <value>	Vault server url (n/a with profile)
--vault-repo <value>	Vault repository (n/a with profile)
--vault-user <value>	Vault user (n/a with profile)
--vault-passwd <value>	Vault password (n/a with profile)
--vault-proxy-domain <value>	Vault proxy domain
--vault-proxy-server <value>	Vault proxy server
--vault-proxy-port <value>	Vault proxy port
--vault-proxy-user <value>	Vault proxy user
--vault-proxy-passwd <value>	Vault proxy password

You must define the Vault server connection to be used. This can either be specified by `vault-server`, `vault-repository`, `vault-user` and `vault-passwd`, or if this information is available in a profile, by using `vault-profile`.

If specified, `vault-profile` will override the other Vault-specific global options

© 2003-2011 SmartBear Software

5.13.2.1 addchanges (for Vault)

```
Usage: ccollab [global-options] addchanges [--diffbranch <value> --upload-
comment <value>] <review> <file-spec> [<file-spec> ...]
```

`addchanges` - Attaches locally-modified files to a review

Command Options

Option	Required?	Description
--diffbranch <value>	N	(ClearCase only) specify either branch-name or "predecessor" to determine the earlier version for upload
--upload-comment <value>	N	Comment used to upload files (default is "Local changes")
<review>	Y	Must be either an integer review-id, 'new', 'ask', or 'last'
<file-spec> [<file-spec> ...]	Y	Local files or directories to scan for modified files

Uploads locally modified files controlled by Vault to the PeerReview Complete server.

The first argument is the review specifier, then any number of files or directories can be given. Files are selected as-is; directories are scanned recursively for locally-modified files, which might include edits, additions, deletions, branches, integrations, moves, copies, and so on.

After the scan is complete, a file list is presented in a graphical editor so you can review the files about to be uploaded and make any corrections you wish.

For example, to create a new review and add all changes in the current directory and below, plus the file foo.txt, you would use:

ccollab addchanges new . foo.txt

To upload modified files from the current working directory and all subdirectories to review 123:

© 2003-2011 SmartBear Software

```
ccollab addchanges 123 .
```

To upload file foo.txt and modified files from c:\dev\project into a brand new review:

```
ccollab addchanges new foo.txt c:\dev\project
```

5.14 Subversion Integration

This section describes PeerReview Complete integration with Subversion:

Subversion Server Integration 397
The PeerReview Complete server can pull committed revisions directly from your Subversion server for review, without users needing to install any client programs.

GUI Client 399
The GUI Client can upload local changes to files 400 in a Working Copy, or upload the files in Revisions 401. The GUI Client can also upload arbitrary Subversion diffs 401, or the difference between two Revisions 402, branches / tags 403, or dates 404.

Command Line Client 405
The Command Line Client can upload local changes to files in a Working Copy, or upload the files in committed Revisions. The Command Line Client can also upload arbitrary Subversion diffs.

Subversion Server Hooks
Subversion server-side hooks can ensure code is reviewed 410 or automatically create reviews 410.

Supported Versions
Our Command Line Client and GUI Client integrations use your own Subversion command-line client (`svn`) to communicate with the server. We support these versions:

- v1.6.x
- v1.5.x
- v1.4.x
- v1.3.2 and earlier, only `ccollab addsvndiffs` 220 is supported

Because we use Subversion client applications already present on your computer, we support all protocols, authentications, proxies, and other client configuration options you are currently using.

Support for Directory-level New/Delete/Copy/Move
PeerReview Complete partially supports Subversion's concept of directories (not just files) being altered.

All files underneath the directories in question will be scanned, uploaded and represented properly in the GUI. The directories themselves will *not* be shown in any GUI, or even in the file list confirmation screen presented by the command-line client.

© 2003-2011 SmartBear Software

This all works correctly even if for example you move a parent directory *and* alter/add/delete a file within that directory. You'll get the correct content for the file but the directory itself will not be listed in the review.

Note this limitation affects "commit" functionality. If a directory is and a file within it are modified and reviewed, the PeerReview Complete client will not be able to commit them automatically. The workaround is to commit the changes manually and then "dismiss" the commit Action Item.

Support for `svn+ssh`

The PeerReview Complete clients support running Subversion over SSH (the `svn+ssh://` protocol). On some systems, especially those under Unix, Mac OS X or Cygwin/Windows, this will probably work correctly out of the box.

If a banner is displayed by your SSH connection, this will interfere with the client being able to correctly parse Subversion output. You should be able to suppress SSH banner output by adding the "-q" option to your SSH executable, i.e.:

```
svn_ssh = ssh -q
```

or by creating a local SSH configuration file and adding the line:

```
LogLevel QUIET
```

For some Windows installations, especially with `TortoiseSVN`, you might have to configure Subversion to default to your SSH client. To do this, go to `APPDATA\Subversion` and edit the `config` file (here `APPDATA` refers to your Profile directory under `Application Data`, so e.g. `C:\Documents and Settings\username\Application Data`). Find the line under the `[tunnels]` section that starts with `"ssh="`. Set this to use your SSH client, which if you use `TortoiseSVN` will look something like:

```
ssh = "C:\\Program Files\\TortoiseSVN\\bin\\TortoisePlink.exe" -
l <your username>
```

Note that those double-slashes are important!

Changes to Properties

PeerReview Complete doesn't specifically support reviews of the changes to Subversion properties.

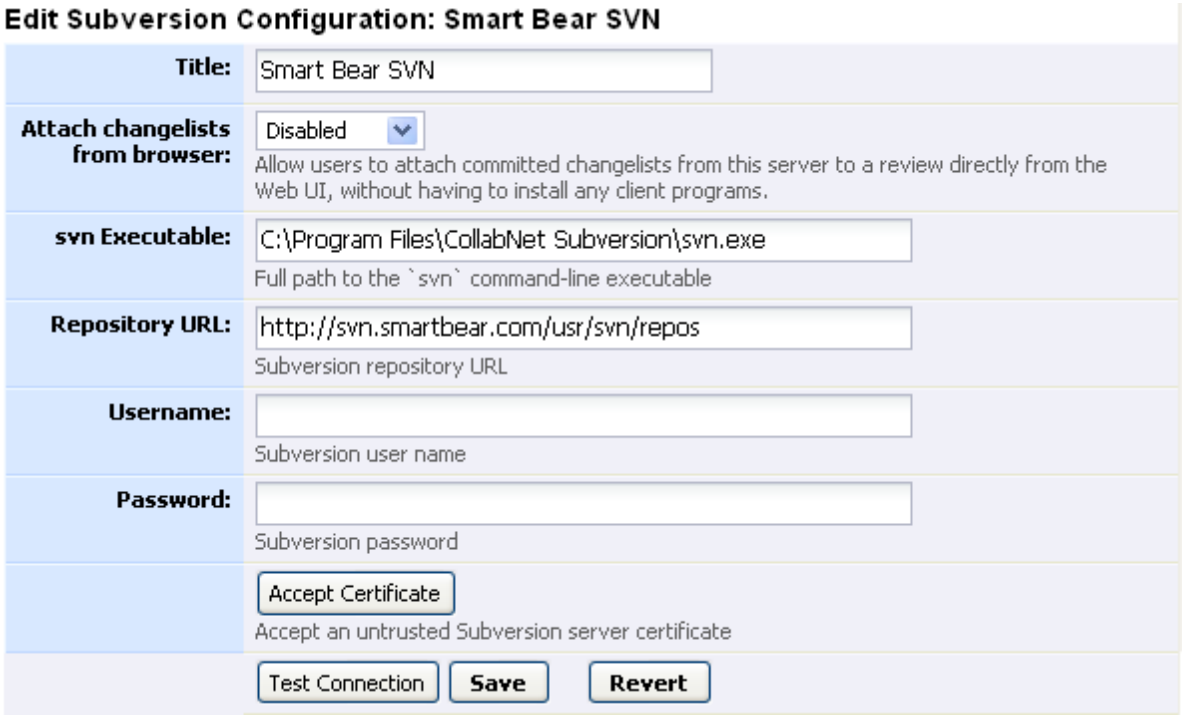
Support for Symlinks

PeerReview Complete partially supports Subversion's ability to version Symlinks. A Symlink contained in a committed Revision uploaded to a Review will be displayed correctly. However a Symlink uploaded to a Review before commit will have incorrect content (the Symlink will be traversed).

5.14.1 Subversion Server Integration

The PeerReview Complete server can be configured to communicate directly with your Subversion server. This allows users to review committed revisions completely from the browser, without having to install any client programs. To enable this feature, create an entry for your Subversion server in the Version Control 101 tab of the administration interface. Version control server entries are also created automatically if one of the client programs uploads files from a server that doesn't match any of the currently configured servers.

© 2003-2011 SmartBear Software



Title The title is displayed to users, so it should be something that everyone will understand, even if they're going through proxies, VPNs, or other such things. When a version control server entry is created automatically, this is filled in with the URL the client used to connect to the Subversion server.

Attach changelists from browser If enabled, this feature lets users select committed revisions to review directly from the web browser, without having to install any client programs.

svn Executable Full path to the 'svn' executable

Repository URL Subversion repository URL. When a version control server entry is created automatically, this is filled in with the URL the client used to connect to the Subversion server.

Username Subversion user name

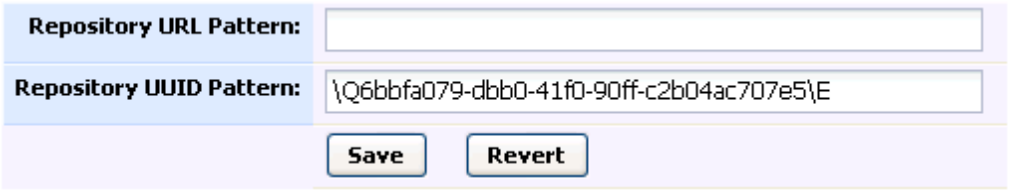
Password Subversion password

Click the ***Test Connection*** to make sure PeerReview Complete can contact your Subversion server. If you get an "untrusted certificate authority" error, this is probably because your Subversion server is using a self-signed certificate. You can click ***Accept Certificate*** to tell PeerReview Complete to permanently accept the certificate.

Client Configuration Mapping

You can supply java-style regular expressions to map changelists uploaded from our client tools to this Subversion server. It is important to set up these regular expressions so that files uploaded by the various PeerReview Complete client tools are correctly associated with this server-side Subversion configuration.

© 2003-2011 SmartBear Software



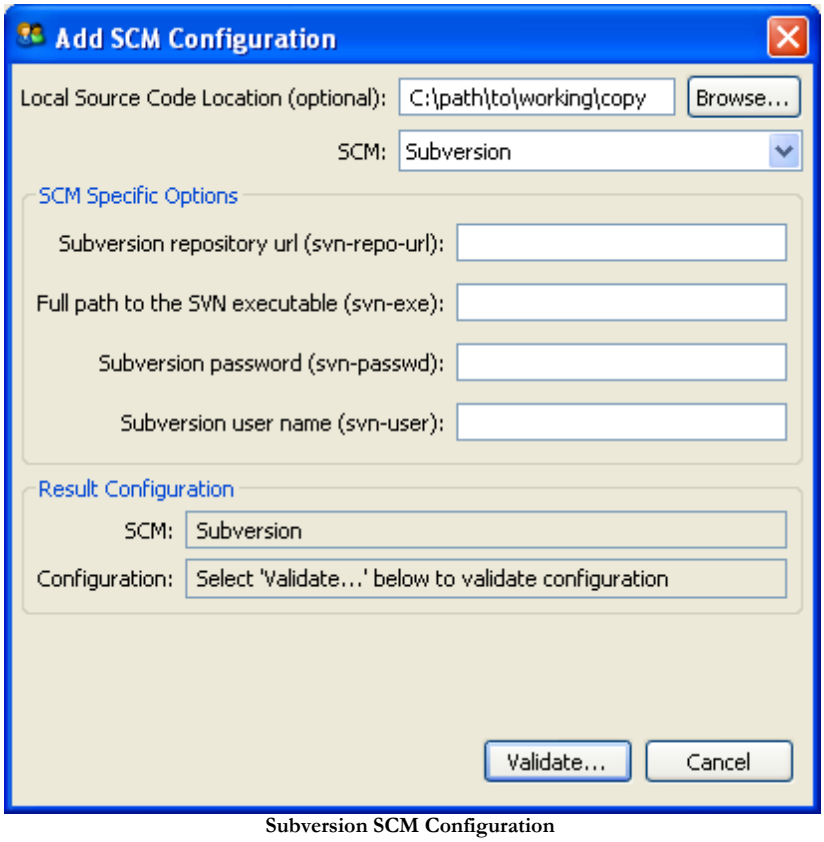
Repository URL Pattern Match on the client's configured repository URL. This is not very reliable because clients may have various network configurations that make the URL look different. It's usually far better to use theRepository UUID Pattern 399.

Repository UUID Pattern Match on the "Repository UUID" returned from running "`svn info`". This is a unique ID generated by every Subversion repository, and usually works perfectly for identifying uploads to this Subversion server. When a version control server entry is created automatically, this is filled in with the "Repository UUID" obtained from the client.

5.14.2 GUI Client

Subversion-specific Options

The SCM Configuration dialog 191 has several Subversion-specific options.



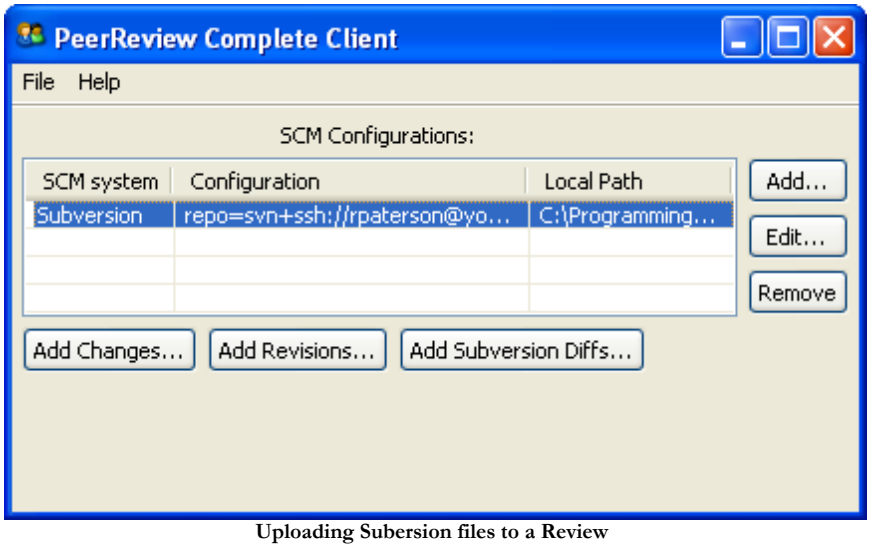
Subversion SCM Configuration

© 2003-2011 SmartBear Software

Uploading files to a Review

Selecting a Subversion SCM Configuration in the GUI Client main screen [190] causes several Add to Review buttons to appear. The *Add Changes...* button will be disabled if you have not specified a working copy in the *Local Path* field of the SCM Configuration dialog [399].

The *Add Changes...* [400] button uploads modified files from a Working Copy. The *Add Revisions...* [401] button uploads files in committed Revisions. The *Add Subversion Diffs...* [401] button uploads arbitrary diffs, or compares the difference between two Revisions [402], branches / tags [403], or dates [404].

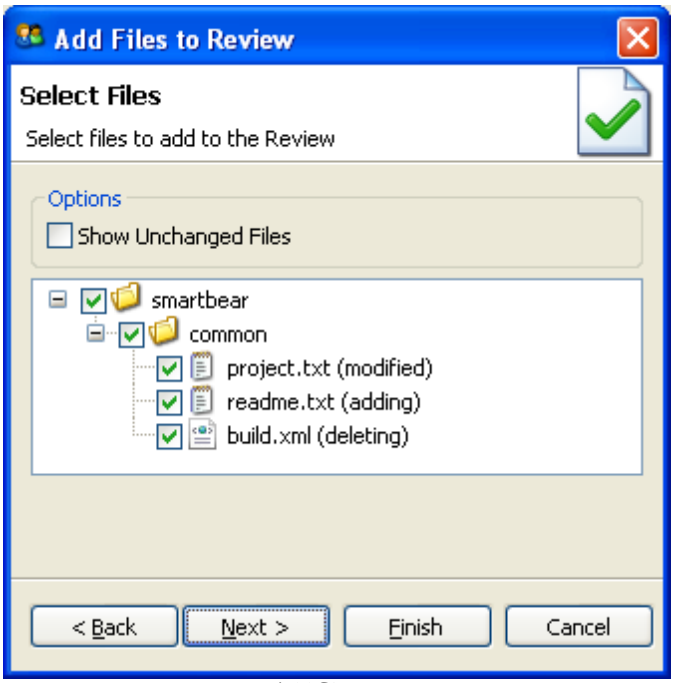


Uploading Subersion files to a Review

Add Changes

Press the *Add Changes...* button to upload modified files in a Subversion working copy to the PeerReview Complete Server for review. The *Add Changes...* button will be disabled if you have not specified a working copy in the *Local Path* field of the SCM Configuration dialog [399].

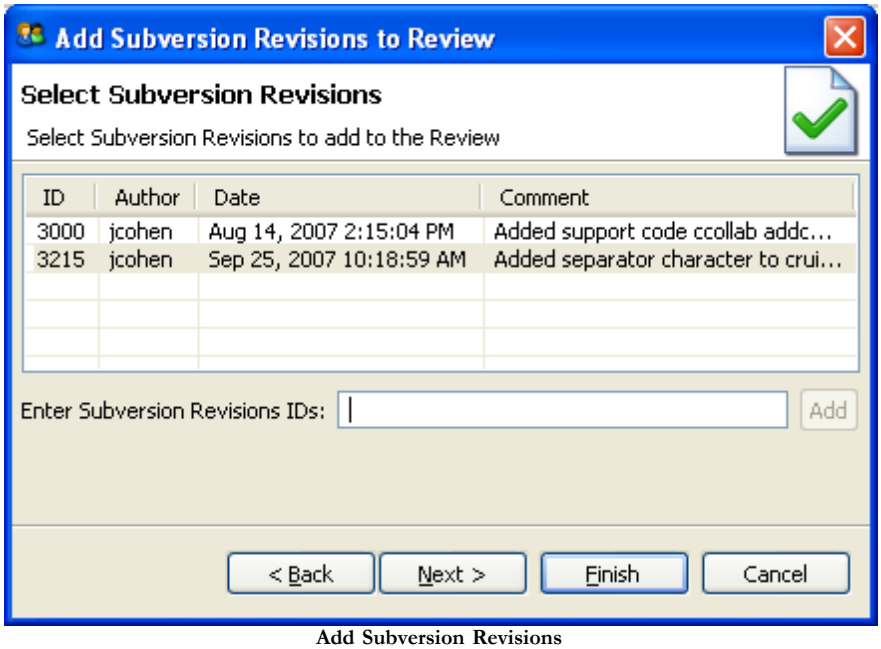
© 2003-2011 SmartBear Software



Add Changes

Add Revisions

Press the *Add Revisions...* button to upload committed Revisions to the PeerReview Complete Server for review. All of the files modified in the selected Revisions will be uploaded.

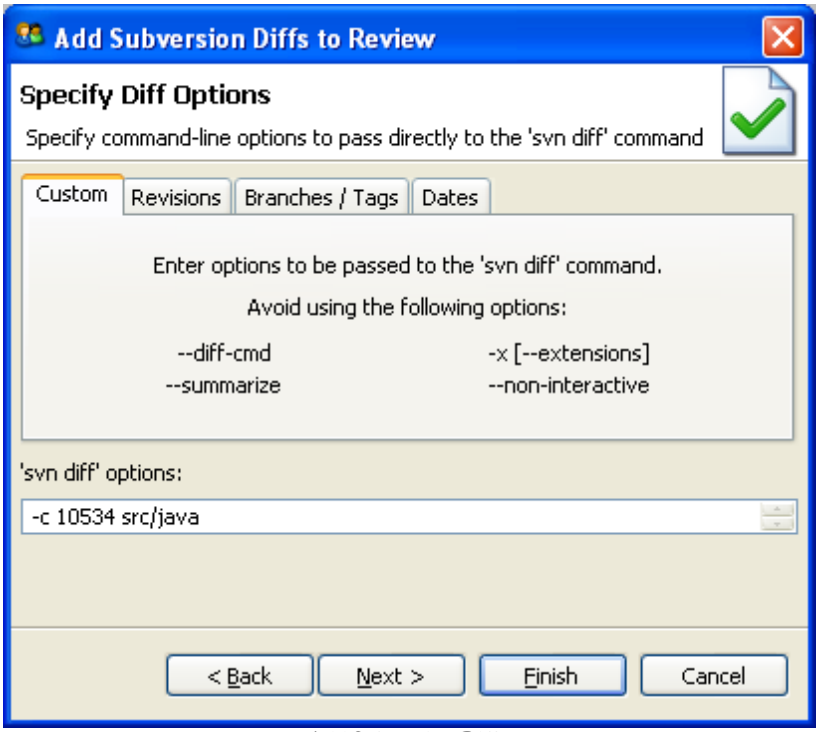


Add Subversion Revisions

Add Subversion Diffs

Press the *Add Subversion Diffs...* button to upload arbitrary Subversion diffs to the PeerReview Complete Server for review.

© 2003-2011 SmartBear Software



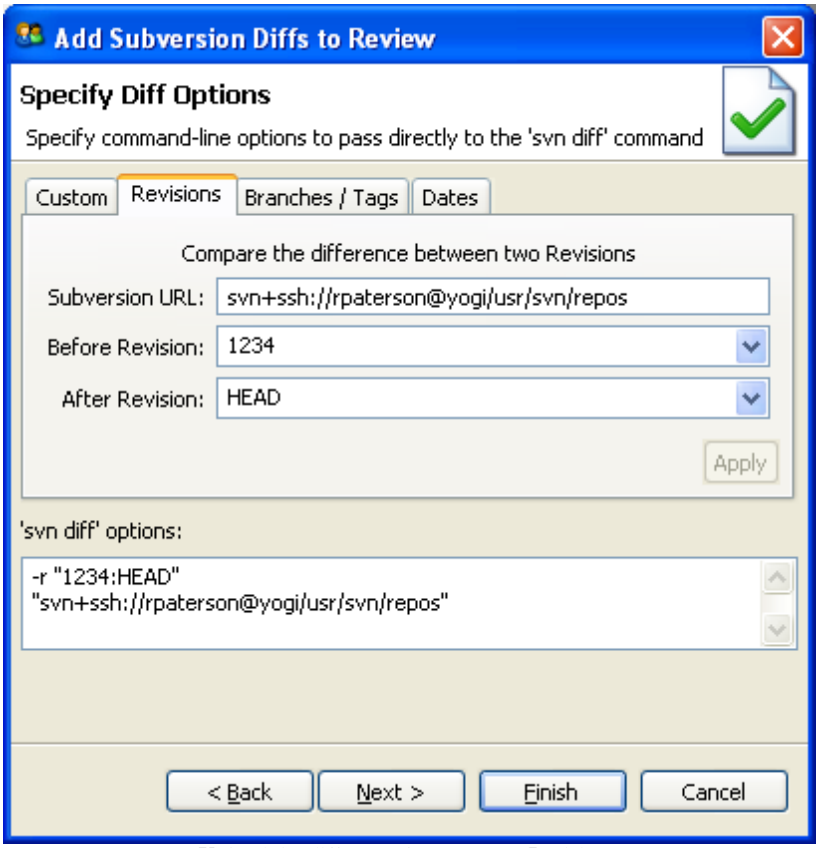
Add Subversion Diffs

You can enter arbitrary Subversion diff options, orcompare the difference between two Revisions 402, branches / tags 403, or dates 404.

5.14.2.1 Comparing two Revisions

Press the *Add Subversion Diffs...* 401 button on the main screen 190 and then select the *Revisions* tab to upload the difference between two Revisions.

© 2003-2011 SmartBear Software

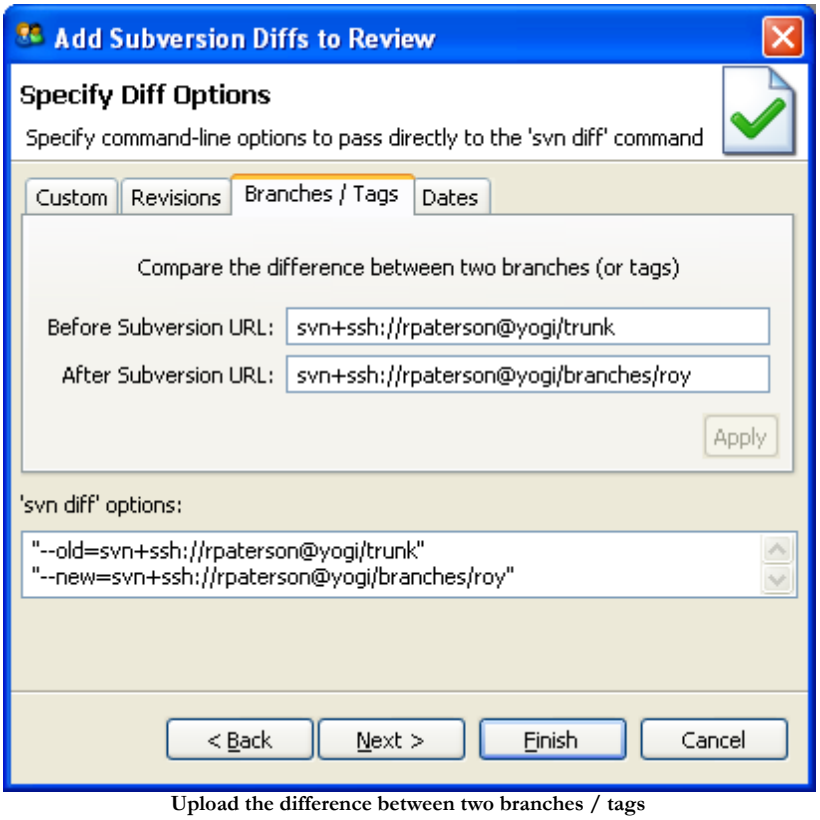


Upload the difference between two Revisions

5.14.2.2 Comparing two branches / tags

Press the *Add Subversion Diffs...* 401 button on the main screen 190 and then select the *Branches* tab to upload the difference between two branches / tags.

© 2003-2011 SmartBear Software

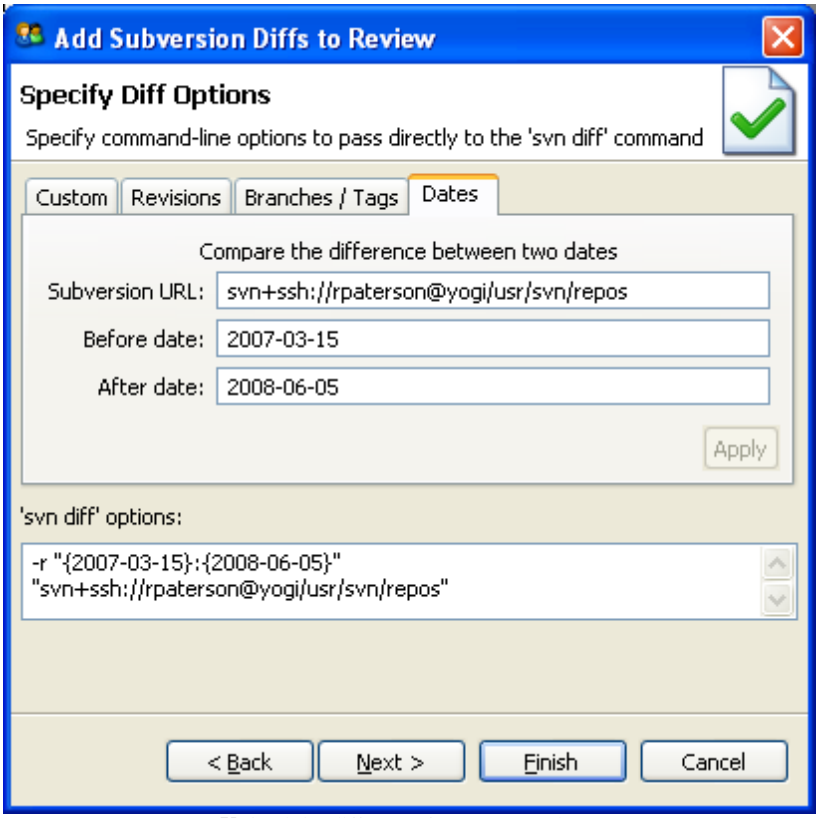


Upload the difference between two branches / tags

5.14.2.3 Comparing two dates

Press the *Add Subversion Diffs...* 401 button on the main screen 190 and then select the *Dates* tab to upload the difference between two dates.

© 2003-2011 SmartBear Software



Upload the difference between two dates

5.14.3 Command Line Client

Commands recommended for Subversion

ccollab addchanges 406 - Attaches locally-modified files to a review

ccollab addchangelist 408 - Attaches an atomic changelist to a review

ccollab addsvndiffs 220 - Uploads diffs generated from the svn diff command

ccollab commit 409 - Commit changes in the review

The addchanges 406 command will upload local changes to files controlled by Subversion before they are checked into version control.

The addchangelist 408 command will upload committed Subversion revisions. The changelist 408 id is the Subversion Revision number. All the files involved in the Revision are uploaded.

Configuration

In most cases, the Command Line Client can automatically detect your Subversion configuration. Try testing your configuration 197 to verify the configuration is detected correctly.

If the Command Line Client is unable to detect your Subversion configuration or you want to override the detected settings, you can manually specify Subversion settings using global options 209.

© 2003-2011 SmartBear Software

To manually configure the Command Line Client to use Subversion, execute the following command:

`ccollab set` 215 `scm` 205 `subversion`

Subversion-specific Options

Option	Description
--svn-exe <value>	Full path to the `svn` command-line executable
--svn-look-exe <value>	Full path to the `svnlook` command-line executable (used by Subversion triggers)
--svn-repo-url <value>	Subversion repository URL
--svn-user <value>	Subversion user name
--svn-passwd <value>	Subversion password
--svn-require-client-certificate-password <value>	Use this if you have a non-empty and unsaved password for your SSL Client Certificate
--svn-auto-add	Treat unversioned files as if they had been added to Subversion
--svn-recurse-externals	Recurse in the 'svn:externals' directories as if they were part of the same repository
--svn-repo-path <value>	Full path to the repository (used by Subversion Triggers)

If you want to review committed Subversion revisions but you don't have a working copy checked out, you must configure your Subversion URL using`svn-repo-url`.

5.14.3.1 addchanges (for Subversion)

```
Usage: ccollab [global-options] addchanges [--diffbranch <value> --upload-
comment <value>] <review> <file-spec> [<file-spec> ...]
```

`addchanges` - Attaches locally-modified files to a review

Command Options

© 2003-2011 SmartBear Software

Option	Required?	Description
--diffbranch <value>	N	(ClearCase only) specify either branch-name or "predecessor" to determine the earlier version for upload
--upload-comment <value>	N	Comment used to upload files (default is "Local changes")
<review>	Y	Must be either an integer review-id, 'new', 'ask', or 'last'
<file-spec> [<file-spec> ...]	Y	Local files or directories to scan for modified files

Uploads locally modified files controlled by Subversion to the PeerReview Complete server.

The first argument is the review specifier, then any number of files or directories can be given. Files are selected as-is; directories are scanned recursively for locally-modified files, which might include edits, additions, deletions, branches, integrations, moves, copies, and so on.

After the scan is complete, a file list is presented in a graphical editor so you can review the files about to be uploaded and make any corrections you wish.

For example, to create a new review and add all changes in the current directory and below, plus the file foo.txt, you would use:

```
ccollab addchanges new . foo.txt
```

To upload modified files from the current working directory and all subdirectories to review 123:

```
ccollab addchanges 123 .
```

To upload file foo.txt and modified files from c:\dev\project into a brand new review:

```
ccollab addchanges new foo.txt c:\dev\project
```

© 2003-2011 SmartBear Software

5.14.3.2 addchangelist (for Subversion)

```
Usage: ccollab [global-options] addchangelist <review> <changelist>
[<changelist> ...]
```

`addchangelist` - Attaches an atomic changelist to a review

Command Options

Option	Required?	Description
<review>	Y	Must be either an integer review-id, 'new', 'ask', or 'last'
<changelist> [<changelist> ...]	Y	SCM-specific ID of an atomic set of changes

Uploads Revisions from Subversion to the PeerReview Complete server.

The first argument is the review specifier, subsequent arguments are the IDs of the Revisions to upload.

For example, to upload Revisions r4321 and r7568 to a new review:

ccollab addchangelist new 4321 7568

For example, to upload Revisions r5432 and r12654 to review 111:

ccollab addchangelist 111 5432 12654

5.14.3.3 addsvndiffs

```
Usage: ccollab [global-options] addsvndiffs [--upload-comment <value>]
<review> [<user-diff-arg> [<user-diff-arg> ...]]
```

`addsvndiffs` - Uploads diffs generated from the svn diff command

Command Options

© 2003-2011 SmartBear Software

Option	Required?	Description
--upload-comment <value>	N	Comment used to upload files (defaults to command-line arguments)
<review>	Y	Must be either an integer review-id, 'new', 'ask', or 'last'
<user-diff-arg> [<user-diff-arg> ...]	N	Command-line arguments to pass directly to the diff command

Upload differences between arbitrary versions of files in Subversion using 'svn diff'.

Do not use diff arguments that affect the diff output such as --diff-cmd, -x [--extensions], --summarize, or --non-interactive. The PeerReview Complete command-line client will automatically select an output format that ensures you will get all the data you need on the server.

Examples:

ccollab addsvndiffs 698 -r 2:16

ccollab addsvndiffs new -r PREV http://my.svn.server/svn/repo

ccollab addsvndiffs 698 -c 10534 src/java

ccollab addsvndiffs new OLDURL[@OLDREV] NEWURL[@NEWREV]

5.14.3.4 commit (for Subversion)

```
Usage: ccollab [global-options] commit [--comment <value> --dismiss-only --
force] <review>
```

`commit` - Commit changes in the review

Command Options

Option	Required?	Description
--comment <value>	N	Comment for reviewed changes
--dismiss-only	N	Just dismiss the Action Item

© 2003-2011 SmartBear Software

Option	Required?	Description
--force	N	Ignore potential problems
<review>	Y	Must be either an integer review-id, 'ask', or 'last'

Commit the changes from a pre-commit review to sourcecontrol. Be sure to include a relevant comment.

5.14.4 Subversion Server Hooks

Triggers recommended for Subversion

ccollab admin trigger ensure-review-started 411 - Changelist cannot be submitted until review of this changelist exists

ccollab admin trigger ensure-reviewed 412 - Changelist cannot be submitted until review of this changelist has been completed

ccollab admin trigger create-review 270 - Creates a new Review for a changelist if no Review already exists

The ensure-review-started 411 and ensure-reviewed 412 hooks ensure that files cannot be committed unless certain conditions are met. If a user attempts to commit a files that doesn't meet those conditions, an error message describing the unfulfilled conditions will be displayed and the files will not be committed. If the conditions are met, the commit will be allowed to continue normally.

The create-review 270 hook automatically creates a review in PeerReview Complete after the revision is committed to the Subversion server. Because you can supply the regular expression for identifying reviews, you can provide users with the ability to review before check-in without having an additional review automatically created after the check-in. This way some groups (or just some check-ins arbitrarily) can use pre-commit review and others post-commit, and either way you know all code has been reviewed or at least that a review of all code exists in the system.

Linking reviews with commits

To use the ensure-review-started 411 and ensure-reviewed 412 hooks, you must first require developers to put the review ID somewhere in the Subversion commit message (also optionally for the create-review 270 hook). The format of this text is completely up to you; you will need to supply a Java-style regular expression that identifies this text and specifically calls out the review ID inside that text. The regular expression is specified using the --review-id-regex hook command option.

Here are some common ways of specifying the review ID and the corresponding regular expressions. Note that regular expressions are *case-insensitive* and you must *identify the review ID portion with parenthesis*:

Text	**--review-id-regex**
`Review: 4233`	`review:\s*(\d+)`

© 2003-2011 SmartBear Software

Text	--review-id-regex
rID4233	rid(\d+)
(review 4233)	\(review (\d+)\)

This text can appear in-line with other text or in a more formal "form-style" layout. Because you control the regular expression, you can control exactly what this looks like.

For more information about Subversion hooks in general, see the Subversion documentation.

5.14.4.1 ensure-review-started (for Subversion)

```
Usage: ccollab [global-options] admin trigger ensure-review-started [--
ignore-integrate --review-id-regex <value>] <changelist-id>
```

`ensure-review-started` - Changelist cannot be submitted until review of this changelist exists

Command Options

Option	Required?	Description
--ignore-integrate	N	Allow integration changes to proceed without review
--review-id-regex <value>	N	Regular Expression that identifies Review ID in commit comment
<changelist-id>	Y	Changelist ID

This trigger ensures that a Review has been started before changes can be committed to version control.

If a Review has not been started, the commit will fail and the user will get an error message explaining that the changes need to be reviewed before commit.

To install this trigger you will need to create a pre-commit hook. If you already have a pre-commit hook, you can add our tool wherever it is appropriate; otherwise you will need to create an executable hook as described in the Subversion documentation (typically a batch file under Windows or a shell script under Linux/Mac).

Example Windows batch file:

© 2003-2011 SmartBear Software

"C:\Program Files\PeerReview Complete Client\ccollab.exe" --url <collabUrl> --user <collabUser> --password <collabPasswd> --scm subversion --svn-repo-path %1 --svn-look-exe "C:\Program Files\Subversion\bin\svnlook.exe" admin trigger ensure-review-started --review-id-regex "review:\s+(\d+)" %2 || exit 1

Example Linux/OSX shell script:

/collab/install/ccollab --url <collabUrl> --user <collabUser> --password <collabPasswd> --scm subversion --svn-repo-path $1 --svn-look-exe /usr/bin/svnlook admin trigger ensure-review-started --review-id-regex "review:\s+(\d+)" $2 || exit 1

Note our use of "exit 1" to ensure that the script terminates with a non-zero exit code if our trigger application fails.

You must specify the --svn-repo-path and --svn-look-exe global options.

You must require developers to put the review ID somewhere in the Subversion commit message. The format of this text is completely up to you; you must supply a Java-style regular expression that identifies this text and specifically calls out the review ID inside that text using the --review-id-regex command option.

5.14.4.2 ensure-reviewed (for Subversion)

```
Usage: ccollab [global-options] admin trigger ensure-reviewed [--ignore-
integrate --review-id-regex <value>] <changelist-id>
```

`ensure-reviewed` - Changelist cannot be submitted until review of this changelist has been completed

Command Options

Option	Required?	Description
--ignore-integrate	N	Allow integration changes to proceed without review
--review-id-regex <value>	N	Regular Expression that identifies Review ID in commit comment
<changelist-id>	Y	Changelist ID

© 2003-2011 SmartBear Software

This trigger ensures that a Review has been completed before changes can be committed to version control.

If a Review has not been completed, the commit will fail and the user will get an error message explaining that the changes need to be reviewed before commit.

To install this trigger you will need to create a pre-commit hook. If you already have a pre-commit hook, you can add our tool wherever it is appropriate; otherwise you will need to create an executable hook as described in the Subversion documentation (typically a batch file under Windows or a shell script under Linux/Mac).

Example Windows batch file:

"C:\Program Files\PeerReview Complete Client\ccollab.exe" --url <collabUrl> --user <collabUser> --password <collabPasswd> --scm subversion --svn-repo-path %1 --svn-look-exe "C:\Program Files\Subversion\bin\svnlook.exe" admin trigger ensure-reviewed --review-id-regex "review:\s+(\d+)" %2 || exit 1

Example Linux/OSX shell script:

/collab/install/ccollab --url <collabUrl> --user <collabUser> --password <collabPasswd> --scm subversion --svn-repo-path $1 --svn-look-exe /usr/bin/svnlook admin trigger ensure-reviewed --review-id-regex "review:\s+(\d+)" $2 || exit 1

Note our use of "exit 1" to ensure that the script terminates with a non-zero exit code if our trigger application fails.

You must specify the --svn-repo-path and --svn-look-exe global options.

You must require developers to put the review ID somewhere in the Subversion commit message. The format of this text is completely up to you; you must supply a Java-style regular expression that identifies this text and specifically calls out the review ID inside that text using the --review-id-regex command option.

5.14.4.3 create-review

```
Usage: ccollab [global-options] admin trigger create-review [--add-on-match
<value> --review-id-regex <value>] <changelist>
```

`create-review` - Creates a new Review for a changelist if no Review already exists

© 2003-2011 SmartBear Software

Command Options

Option	Required?	Description
--add-on-match <value>	N	If a review ID regex matches, add this changelist to the review
--review-id-regex <value>	N	Regular Expression that identifies Review ID in commit comment
<changelist>	Y	SCM-specific ID of an atomic set of changes

The create-review trigger automatically creates a review in PeerReview Complete after the change is committed to the Subversion server. It's smart enough to not create reviews if they've already been created for this code.

For this hook to work smoothly you will need your Subversion usernames and PeerReview Complete logins to match. Here "match" means "differs at most in capitalization." This ensures that when a developer checks in code the review is created under his PeerReview Complete account.

If you don't do this the hook will still work, but reviews will be created under the system administrator's account. Someone will have to log into the system with that account and assign the review to someone else.

You may optionally specify a regular expression that identifies a PeerReview Complete review ID inside a Subversion commit message. If you supply the regular expression, the create-review hook will check for this text in commit messages and automatically not create a new review if a non-canceled review with that ID already exists. This allows some users to do pre-commit review and others post-commit, and either way you know all code has been reviewed or at least that a review of all code exists in the system.

For Subversion, to install this trigger you will need to create a post-commit hook. If you already have a post-commit hook, you can add our tool wherever it is appropriate; otherwise you will need to create an executable hook as described in the Subversion documentation (typically a batch file under Windows or a shell script under Linux/Mac).

Example Windows batch file:

"C:\Program Files\PeerReview Complete Client\ccollab.exe" --url <collabUrl> --user <collabUser> --password <collabPasswd> --scm subversion --svn-repo-url svn://url/to/repo --svn-exe "C:\Program Files\Subversion\bin\svn.exe" admin trigger create-review --review-id-regex "review:\s+(\d+)" %2 || exit 1

Example Linux/OSX shell script:

© 2003-2011 SmartBear Software

/collab/install/ccollab --url <collabUrl> --user <collabUser> --password <collabPasswd> --scm subversion --svn-repo-url svn://url/to/repo --svn-exe /usr/bin/svn admin trigger create-review --review-id-regex "review:\s+(\d+)" $2 || exit 1

Note our use of "exit 1" to ensure that the script terminates with a non-zero exit code if our trigger application fails.

5.14.4.4 ensure-review-started (for Subversion)

```
Usage: ccollab [global-options] admin trigger ensure-review-started [--
ignore-integrate --review-id-regex <value>] <changelist-id>
```

`ensure-review-started` - Changelist cannot be submitted until review of this changelist exists

Command Options

Option	Required?	Description
--ignore-integrate	N	Allow integration changes to proceed without review
--review-id-regex <value>	N	Regular Expression that identifies Review ID in commit comment
<changelist-id>	Y	Changelist ID

This trigger ensures that a Review has been started before changes can be committed to version control.

If a Review has not been started, the commit will fail and the user will get an error message explaining that the changes need to be reviewed before commit.

To install this trigger you will need to create a pre-commit hook. If you already have a pre-commit hook, you can add our tool wherever it is appropriate; otherwise you will need to create an executable hook as described in the Subversion documentation (typically a batch file under Windows or a shell script under Linux/Mac).

Example Windows batch file:

© 2003-2011 SmartBear Software

"C:\Program Files\PeerReview Complete Client\ccollab.exe" --url <collabUrl> --user <collabUser> --password <collabPasswd> --scm subversion --svn-repo-path %1 --svn-look-exe "C:\Program Files\Subversion\bin\svnlook.exe" admin trigger ensure-review-started --review-id-regex "review:\s+(\d+)" %2 || exit 1

Example Linux/OSX shell script:

/collab/install/ccollab --url <collabUrl> --user <collabUser> --password <collabPasswd> --scm subversion --svn-repo-path $1 --svn-look-exe /usr/bin/svnlook admin trigger ensure-review-started --review-id-regex "review:\s+(\d+)" $2 || exit 1

Note our use of "exit 1" to ensure that the script terminates with a non-zero exit code if our trigger application fails.

You must specify the --svn-repo-path and --svn-look-exe global options.

You must require developers to put the review ID somewhere in the Subversion commit message. The format of this text is completely up to you; you must supply a Java-style regular expression that identifies this text and specifically calls out the review ID inside that text using the --review-id-regex command option.

5.14.4.5 ensure-reviewed (for Subversion)

```
Usage: ccollab [global-options] admin trigger ensure-reviewed [--ignore-
integrate --review-id-regex <value>] <changelist-id>
```

`ensure-reviewed` - Changelist cannot be submitted until review of this changelist has been completed

Command Options

Option	Required?	Description
--ignore-integrate	N	Allow integration changes to proceed without review
--review-id-regex <value>	N	Regular Expression that identifies Review ID in commit comment
<changelist-id>	Y	Changelist ID

© 2003-2011 SmartBear Software

This trigger ensures that a Review has been completed before changes can be committed to version control.

If a Review has not been completed, the commit will fail and the user will get an error message explaining that the changes need to be reviewed before commit.

To install this trigger you will need to create a pre-commit hook. If you already have a pre-commit hook, you can add our tool wherever it is appropriate; otherwise you will need to create an executable hook as described in the Subversion documentation (typically a batch file under Windows or a shell script under Linux/Mac).

Example Windows batch file:

"C:\Program Files\PeerReview Complete Client\ccollab.exe" --url <collabUrl> --user <collabUser> --password <collabPasswd> --scm subversion --svn-repo-path %1 --svn-look-exe "C:\Program Files\Subversion\bin\svnlook.exe" admin trigger ensure-reviewed --review-id-regex "review:\s+(\d+)" %2 || exit 1

Example Linux/OSX shell script:

/collab/install/ccollab --url <collabUrl> --user <collabUser> --password <collabPasswd> --scm subversion --svn-repo-path $1 --svn-look-exe /usr/bin/svnlook admin trigger ensure-reviewed --review-id-regex "review:\s+(\d+)" $2 || exit 1

Note our use of "exit 1" to ensure that the script terminates with a non-zero exit code if our trigger application fails.

You must specify the --svn-repo-path and --svn-look-exe global options.

You must require developers to put the review ID somewhere in the Subversion commit message. The format of this text is completely up to you; you must supply a Java-style regular expression that identifies this text and specifically calls out the review ID inside that text using the --review-id-regex command option.

5.14.4.6 create-review

```
Usage: ccollab [global-options] admin trigger create-review [--add-on-match
<value> --review-id-regex <value>] <changelist>
```

`create-review` - Creates a new Review for a changelist if no Review already exists

© 2003-2011 SmartBear Software

Command Options

Option	Required?	Description
--add-on-match <value>	N	If a review ID regex matches, add this changelist to the review
--review-id-regex <value>	N	Regular Expression that identifies Review ID in commit comment
<changelist>	Y	SCM-specific ID of an atomic set of changes

The create-review trigger automatically creates a review in PeerReview Complete after the change is committed to the Subversion server. It's smart enough to not create reviews if they've already been created for this code.

For this hook to work smoothly you will need your Subversion usernames and PeerReview Complete logins to match. Here "match" means "differs at most in capitalization." This ensures that when a developer checks in code the review is created under his PeerReview Complete account.

If you don't do this the hook will still work, but reviews will be created under the system administrator's account. Someone will have to log into the system with that account and assign the review to someone else.

You may optionally specify a regular expression that identifies a PeerReview Complete review ID inside a Subversion commit message. If you supply the regular expression, the create-review hook will check for this text in commit messages and automatically not create a new review if a non-canceled review with that ID already exists. This allows some users to do pre-commit review and others post-commit, and either way you know all code has been reviewed or at least that a review of all code exists in the system.

For Subversion, to install this trigger you will need to create a post-commit hook. If you already have a post-commit hook, you can add our tool wherever it is appropriate; otherwise you will need to create an executable hook as described in the Subversion documentation (typically a batch file under Windows or a shell script under Linux/Mac).

Example Windows batch file:

"C:\Program Files\PeerReview Complete Client\ccollab.exe" --url <collabUrl> --user <collabUser> --password <collabPasswd> --scm subversion --svn-repo-url svn://url/to/repo --svn-exe "C:\Program Files\Subversion\bin\svn.exe" admin trigger create-review --review-id-regex "review:\s+(\d+)" %2 || exit 1

Example Linux/OSX shell script:

© 2003-2011 SmartBear Software

/collab/install/ccollab --url <collabUrl> --user <collabUser> --password <collabPasswd> --scm subversion --svn-repo-url svn://url/to/repo --svn-exe /usr/bin/svn admin trigger create-review --review-id-regex "review:\s+(\d+)" $2 || exit 1

Note our use of "exit 1" to ensure that the script terminates with a non-zero exit code if our trigger application fails.

5.15 Surround SCM Integration

This section describes PeerReview Complete integration with Surround SCM.

GUI Client 419
The GUI Client can upload local changes to files in a Surround SCM working directory. The GUI Client can also upload the files in pending or committed changelists.

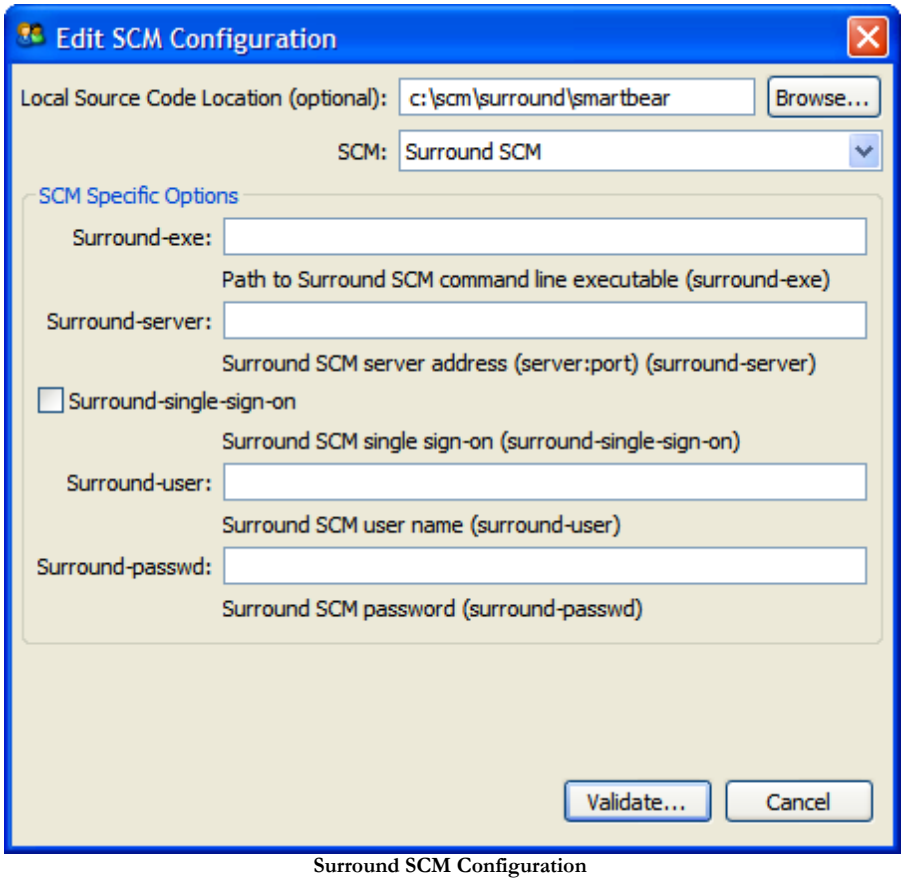
Command Line Client 421
The Command Line Client can upload local changes to files from a Surround SCM working directory before they are checked in, or upload already committed files. The Command Line Clientcan also upload the files in pending or committed changelists.

Supported Versions
Our integration uses your own installed Surround command line client executable (`sscm`) to generate differences for review. We require Surround SCM client version 2009.1.0 or later. Because we use client applications already present on your computer, we support all protocols, authentications, proxies, and other client configuration options you are currently using.

5.15.1 GUI Client

Surround-specific Options
The SCM Configuration dialog 191 has several Surround-specific options. These can be set as necessary to override Surround options derived from the environment.

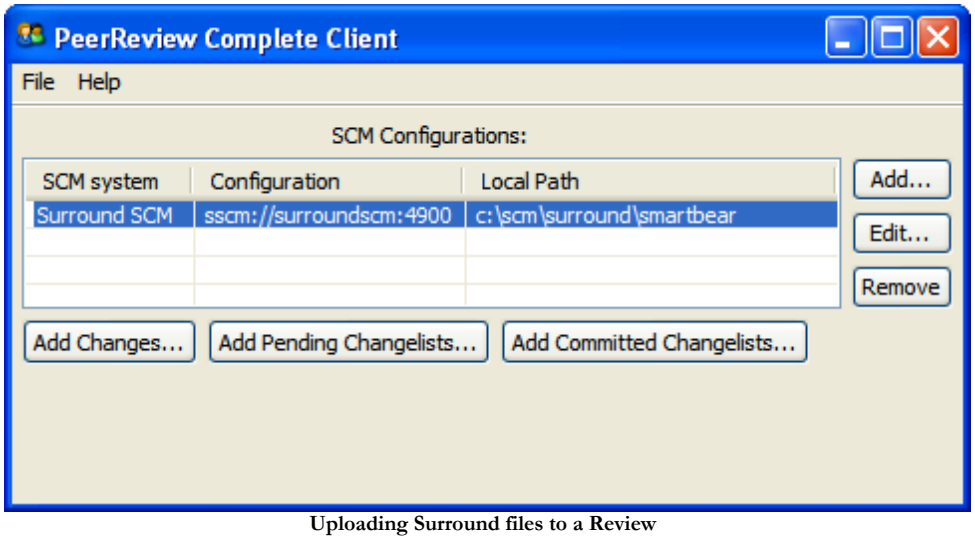
© 2003-2011 SmartBear Software



Surround SCM Configuration

Uploading files to a Review

Selecting a Surround SCM Configuration in the GUI Clientmain screen 190 causes several Add to Review buttons to appear. The *Add Pending Changelists...* 421 button uploads pending changelists, and e*Add Committed Changelists...* 421 button uploads committed changelists.

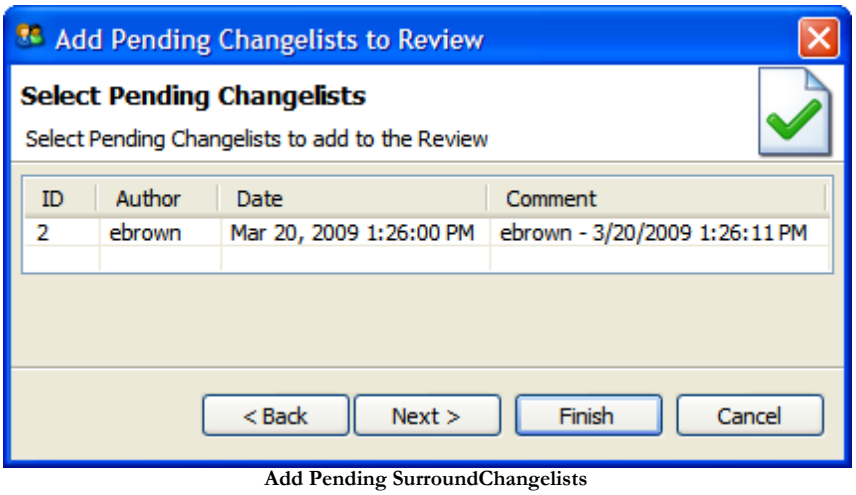


Uploading Surround files to a Review

© 2003-2011 SmartBear Software

Add Pending Surround Changelists

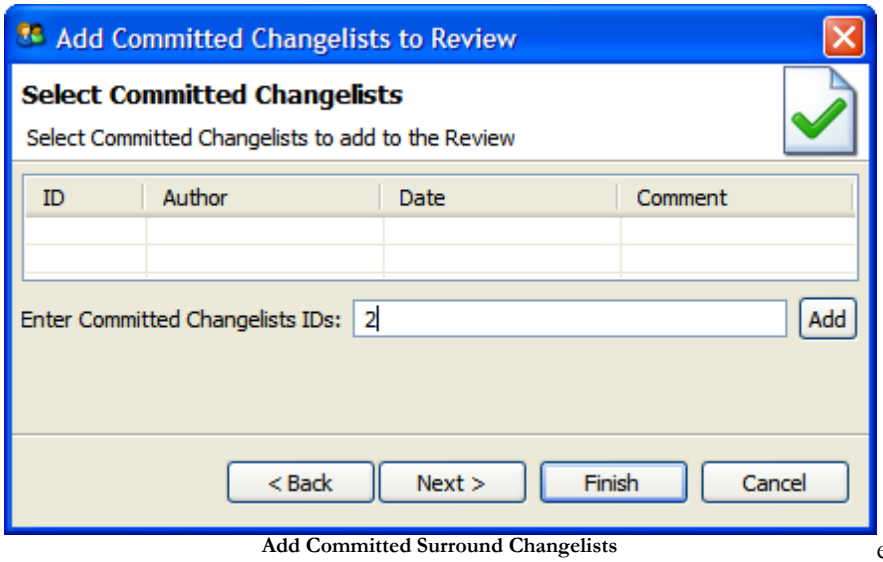
Press the *Add Pending Changelists...* button to upload the files in a pending Surround changelist to the PeerReview Complete Server for review.



Add Pending SurroundChangelists

Add Committed Surround Changelists

Press the *Add Committed Changelists...* button to upload the files in a committed Surround changelist to the PeerReview Complete Server for review. Select the changelist to be added, and press the *Add* button to include it in the review.



Add Committed Surround Changelists

ew.

5.15.2 Command Line Client

Commands recommended for Surround SCM

ccollab addchanges 422 - Attaches locally-modified files to a review

ccollab addchangelist 423 - Attaches an atomic changelist to a review

ccollab addversions 424 - Attaches any 2 given versions to a review

© 2003-2011 SmartBear Software

The addchangelist 423 command will upload pending Surround changelists into PeerReview Complete. Currently, committed changelists are not supported.

Configuration
The Command Line Client can typically detect some of your Surround configuration. Try testing your configuration 197 to verify the configuration is detected correctly.

If the Command Line Client is unable to detect your Surround configuration or you want to override the detected settings, you can manually specify your Surround settings using global options 209.

To manually configure the Command Line Client to use Surround, execute the following command:

`ccollab set` 215 `scm` 205 `surround`

Surround SCM-specific Options

Option	Description
--surround-exe <value>	Full path to the Surround SCM `sscm` command-line client
--surround-server <value>	Surround SCM server address (server:port)
--surround-single-sign-on	Surround SCM single sign-on
--surround-user <value>	Surround SCM user name
--surround-passwd <value>	Surround SCM password

5.15.2.1 addchanges (for Surround SCM)

```
Usage: ccollab [global-options] addchanges [--diffbranch <value> --upload-
comment <value>] <review> <file-spec> [<file-spec> ...]
```

`addchanges` - Attaches locally-modified files to a review

Command Options

Option	Required?	Description
--diffbranch <value>	N	(ClearCase only) specify either branch-name or "predecessor" to determine the earlier version for upload

© 2003-2011 SmartBear Software

Option	Required?	Description
--upload-comment <value>	N	Comment used to upload files (default is "Local changes")
<review>	Y	Must be either an integer review-id, 'new', 'ask', or 'last'
<file-spec> [<file-spec> ...]	Y	Local files or directories to scan for modified files

Uploads locally modified files controlled by Surround SCM to the PeerReview Complete server.

The first argument is the review specifier, then any number of files or directories can be given. Files are selected as-is; directories are scanned recursively for locally-modified files, which might include edits, additions, deletions, branches, integrations, moves, copies, and so on.

After the scan is complete, a file list is presented in a graphical editor so you can review the files about to be uploaded and make any corrections you wish.

For example, to create a new review and add all changes in the current directory and below, plus the file foo.txt, you would use:

```
ccollab addchanges new . foo.txt
```

To upload modified files from the current working directory and all subdirectories to review 123:

```
ccollab addchanges 123 .
```

To upload file foo.txt and modified files from c:\dev\project into a brand new review:

```
ccollab addchanges new foo.txt c:\dev\project
```

5.15.2.2 addchangelist (for Surround SCM)

```
Usage: ccollab [global-options] addchangelist <review> <changelist>
[<changelist> ...]
```

© 2003-2011 SmartBear Software

`addchangelist` - Attaches an atomic changelist to a review

Command Options

Option	Required?	Description
<review>	Y	Must be either an integer review-id, 'new', 'ask', or 'last'
<changelist> [<changelist> ...]	Y	SCM-specific ID of an atomic set of changes

Uploads Pending Changelists or Committed Changelists from Surround SCM to the PeerReview Complete server.

The first argument is the review specifier, subsequent arguments are the IDs of the Pending Changelists or Committed Changelists to upload.

For example, to upload Pending Changelists 4321 and 7568 to a new review:

```
ccollab addchangelist new 4321 7568
```

For example, to upload Committed Changelists 5432 and 12654 to review 111:

```
ccollab addchangelist 111 5432 12654
```

5.15.2.3 addversions (for Surround SCM)

```
Usage: ccollab [global-options] addversions [--upload-comment <value>]
<review> [<version-spec> [<version-spec> ...]]
```

`addversions` - Attaches any 2 given versions to a review

Command Options

© 2003-2011 SmartBear Software

Option	Required?	Description
--upload-comment <value>	N	Comment used to upload files (default is "Local changes")
<review>	Y	Must be either an integer review-id, 'new', 'ask', or 'last'
<version-spec> [<version-spec> ...]	N	Versions to be reviewed are given by <path> <version> [<previous version>], where <path> is the filename or server path of the file, <version> is the version to be reviewed, and <previous version> is an optional version that <version> should be diff'ed against. If <previous version> is not specified, the predecessor version of <version> will be used. If no <version-spec> is given on the command line, then one or more version specs will be read from stdin, one <version-spec> per line as described above. Note that arguments containing spaces must be quoted to be parsed correctly

Uploads specific versions of a file or files to a review. A version-spec is given by a filename followed by a version number, and optionally, a previous version number:

file-name version-number [previous-version-number]

If no previous version number is explicity given, diffs will be generated with the predecessor version as determined from the SCM system.

Adding versions for a single file can be done with command line arguments given after the review number. To upload versions for multiple files, the file and versions may be submitted on lines read from standard input. If

© 2003-2011 SmartBear Software

the path and versions are not given on the command line, the client will expect to read them from stdin. Each line read should be in the format shown above. Some example command lines for reading versions from stdin:

```
ccollab addversions 86753
ccollab addversions last < versionlist.txt
cat versionlist.txt | ccollab addversions new
```

When supplying the version list in the command line client editor or an input file, the keyword 'local' can be used to denote the version corresponding to the local version of the file. The 'local' keyword can only be used for the first version argument, not the previous version.

6 External Integrations

Integrating PeerReview Complete into other systems, scripts, and processes is easy. A wide variety of integration mechanisms is available.

Reporting-writing tools 427
PeerReview Complete comes with a few built-in reports 176, but almost everyone wants to make reports of their own.

You can use any external reporting system including Excel, Access, Crystal Reports, and Business Objects.

Bug-tracking systems 442

PeerReview Complete can integrate with external issue-tracking systems such as TeamTrack, Bugzilla, JIRA, and FogBugz. Integration can be donein several ways depending on the type of integration you require.

Scripting 443
Scripting is the way to implement custom behaviors in PeerReview Complete. You can run scripts on your user's machines, or from the server machine, possible invoked with a trigger. Almost any action you can do in the web interface can be done from a script.

6.1 Custom Reports

PeerReview Complete comes with a number of pre-built and customizable reports 176. However, if you would like to automate a report for integration with external tools or applications, or if you would like to create your own reports, you can access the underlying database directly.

(Note: SmartBear does not support querying Hypersonic 25 databases. You will need to use one of the other databases 24 supported by PeerReview Complete.)

© 2003-2011 SmartBear Software

Creating Custom SQL Reports
In many cases, you may want report output slightly different than one of the existing customizable reports. In these cases, it may be simplest to begin with a report that outputs most of the data that you want, then modify its database query as needed.

1. Open the custom report [176] that best matches the type of report you would like to create.
2. Choose the columns you wish to display in the report, and the criteria by which you would like to filter the results.
3. Click the "Save" button.
4. Verify that the first few results contain the data you expect.
5. Click the "SQL" link at the top-right of the Results pane. This will download a SQL query appropriate for your database which you can use as a starting point for your custom report.
6. Reference the PeerReview Complete database schema [427] and documentation for your database's SQL dialect (MySQL, Oracle, or Microsoft SQL) to modify the query as you see fit.
7. Execute the query directly against the database using whatever SQL tool you like. Some tools that can run SQL queries include: Excel, Access, Crystal Reports, and Business Objects.

Custom XSL Reports
You can alternatively use the Command Line Client to retrieve XML data regarding a review. For samples of this, see: Example XPath and XSL [441].

6.1.1 Database Schema

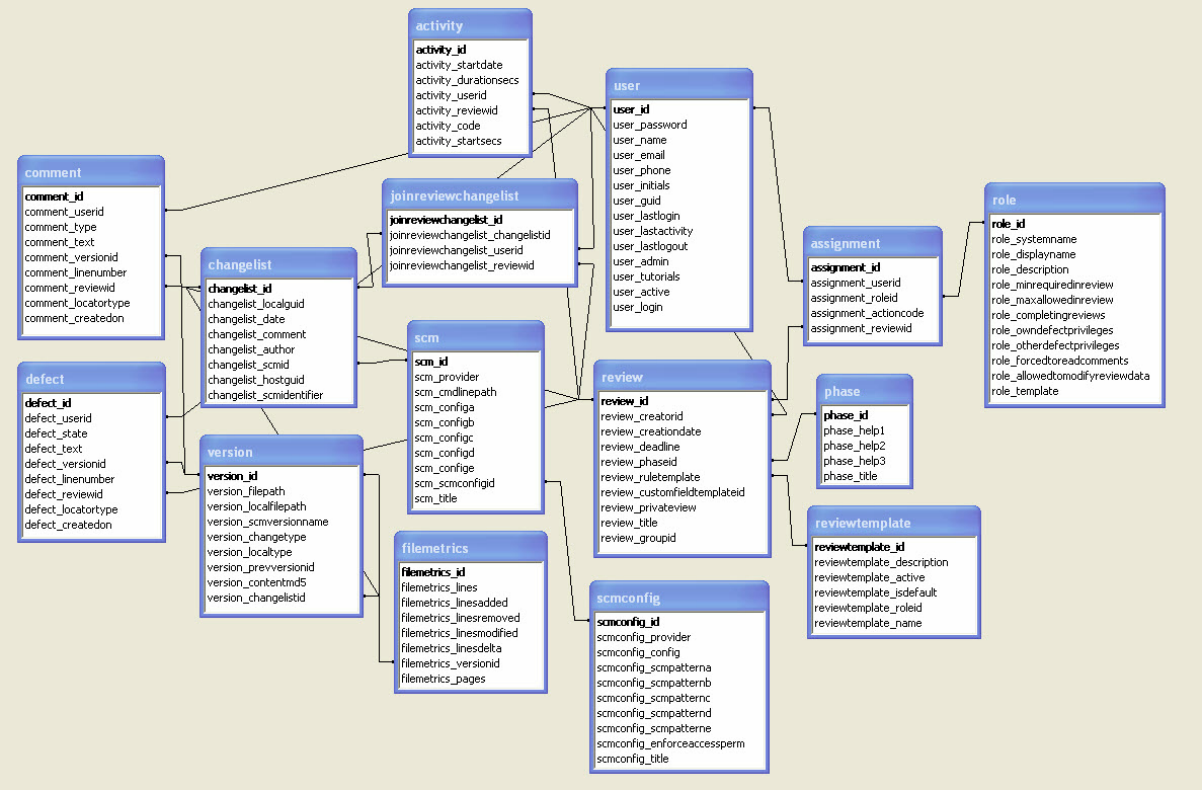
This section covers the PeerReview Complete database schema and some special features of the database we created specifically to support external custom reporting applications.

The Database is Read-Only!
Never change data in the database directly. Although we support read-only access to the database for reporting and automation, if you make changes to data in the database yourself **you could irreparably destroy the integrity of the database.**

Database Schema
The diagram below shows *only the major tables* in the PeerReview Complete database and which files should be linked with which other fields when creating queries.

There are more tables which are either (a) purposefully undocumented or (b) described below but which don't have relationships to other tables. The diagram doesn't contain those tables in the interest of simplicity:

© 2003-2011 SmartBear Software



Database Schema

> **Warning:** SmartBear reserves the right to change any of the table definitions whenever there is a minor point-release of the product.
>
> Note: The table/view names above are approximations. In some cases, the names may be altered slightly (e.g. vowels are removed).
>
> We are committed to backwards-compatibility with the reporting views 432, and with our own client software 196.

Here's a brief description of each table:

`activity` Activity log of users' actions during review. This is used to compute metrics such as inspection rate.

Each row represents a slice of time where the user was active. There might be many slices for a given user in a single review. Each slice includes a duration (in seconds) and a "start time" that is encoded both as a database date/time (`activity_startdate`) and as a number of seconds since 1970-01-01 00:00:00 GMT (`activity_startsecs`).

© 2003-2011 SmartBear Software

The activity action code (`activity_code`) tells whether the user was acting in the capacity of an author (A), reviewer (R), or was doing rework (F) as opposed to actually reviewing. Also there's a code for when someone views the review but is not a participant (P).

It almost always best to use the reporting views [432] to access activity data so you don't get your query wrong. The technique and motivation behind this system is described in our metrics section [455].

`assignment` List of user <--> review combinations. For each user who was "assigned" to a review, notes the role associated with that user.

The `assignment_actioncode` field is used internally to understand what general status that user has in the review currently. You should not depend on this field as we might change it in the future. Refer to the `assignment_state` view for names and descriptions for these codes.

`changelist` Every time a set of files is uploaded to the server a "changelist" record is created. Most changelists will be associated with a review through `joinreviewchangelist`, but some may not be for various reasons. Changelists are linked to `version` where the actual file content is represented.

If the changelist is of files from a local hard drive, the "local GUID" field will contain a globally-unique made-up identifier for that upload. If the changelist was derived from something already checked into version control (e.g. a Perforce or Subversion changelist), then this field will be blank.

The date, author, and check-in comment are all listed if known. To the extent possible this will match usernames with version control.

Also linked is the SCM table ID. This ties the changelist to a specific version control server. Changelists from different servers might match in other details but are actually unrelated.

If the changelist happened to have an associated identifier in a version control system, that is also recorded. Otherwise that field is blank.

`comment` Represents a comment made by a user in some conversation. This includes not only actual chat but indirect events such as "marked read" and "created defect" and "new file uploaded."

Also included is the file (`version`) and line number the comment is associated with, however both of those fields are optional. They are linked to user ID and review ID as well.

`defect` Represents a defect made by a user in some conversation.

Also included is the file (`version`) and line number the defect is associated with, however both of those fields are optional. They are linked to user ID and review ID as well.

A state field indicates whether the defect is still open or fixed. We will be adding more state to this field in the future.

© 2003-2011 SmartBear Software

`filemetrics`	Holds basic change metrics for file versions. Each `filemetrics` record is tied to one `version` record. You should only depend on the values of this table for versions directly associated with changelists. The other metrics are often incomplete. There are technical reasons for this; we will not be changing this behavior.
`groupdescription`	Represents a group 88. Some of the groups are defined in the web UI and some are built-in internal groups automatically defined by PeerReview Complete
`groupusers`	Joins the `groupdescription` table with the `user` table to represent the users that are direct members of a group. A user can be a member of zero, one, or many groups.
`groupgroups`	Joins the `groupdescription` table with itself to represent the groups that are direct members of a group. A group can be a member of zero, one, or many groups.
`groupancestry`	*The behavior of this table is intentionally undocumented.*
`joinreviewchangelist`	Joins reviews and changelists. A changelist can be associated with zero, one, or many reviews, and a review can be associated with zero, one, or many changelists.

`metadatadescription`
`metadatavaluechar acter`
`metadatavaluedate`
`metadatavalueinteger`
`metadatavaluestring`
`metadatavaluestringbig`
	All of these tables have to do with "meta-data" which means any data where the data schema itself is dynamic. Most notably, all review and defect custom fields 104 are a kind of meta-data.
	
	You should not use the meta-data tables directly Their relationships are very complex and we change how they work regularly as we add more features.
	
	Instead, access meta-data through the reporting views described below 432. This contains all the information you need for custom fields and formats it nicely as an added bonus.

`metadataselectitem`	Information for all the drop-down items in any custom field. Each item is matched to a particular custom field. The "title" is the text displayed to end users. A "sequence" number defines the order of the elements (the IDs are not an order). Items can also be individually enabled or disabled.
`notification`	Holds the history of notification messages that have been sent out to clients. Clients might choose (or not) to get notifications by email, RSS feed, etc..

© 2003-2011 SmartBear Software

This table is periodically cleaned out by the server. There will always be some backlog of events for each user (e.g. for use in creating the RSS feed for a user), but you cannot depend on any particular number of events to be saved.

`reportcategory` *Internal server use. Do not depend on this table.*

`reportfilter`

`reporttemplate`

`review` Holds one record for each review in the system.

The "creator" is the user who created the review, or the system administrator if the review was created automatically.

Use the review custom field view 432 to access review custom field data.

`reviewtemplate` *Internal server use. Do not depend on this table.*

`role` Represents all of the roles from all role-sets. Each role has a "standard" name that never changes and the custom name that was set by the user.

`scm` Contains one record for each SCM server that has ever been reported by a client. It's OK if there are duplicate records for a given SCM system. This separates changelists from different systems. This table will probably change in the future.

`user` One record for each user who can log into the system. User ID 1 is the special system administrator.

Key user information and preferences are stored here. Additional user preference information is stored as meta-data and is accessible from the special `userprefs` view.

The `user_initials` field is deprecated and should be ignored.

Passwords are stored in hashed form so that a casual observer cannot deduce a password. If you need to reset a password, set this field to:

`d41d8cd98f00b204e9800998ecf8427e`

`version` One record for every version of every file that has ever been uploaded to the server. Join with `changelist` to see the group they were uploaded with.

Each version includes the full file path to the original document. If this file was retrieved from version control, this will be the version control server path, not the path on the user's local hard drive.

The version name is the version control-specific name of the version of the file. This i typically a number or set of numbers.

© 2003-2011 SmartBear Software

The version change-type indicates whether this represents a file addition, deletion, modification, and so on. Sometimes the system doesn't know. You should use this as guide but not depend on it because there are exceptions to the "type" rules and we add new types periodically. The current values are:

```
? - Unknown
A - Added
B - Branched
D - Deleted
I - Integrated
M - Modified
R - Reverted
U - Uploaded
```

Sometimes the version will have a "previous" version. This typically means the version that came before it in version control. This is used internally and is tricky; there are lot of exceptions and we can change exactly what this means.

The content MD5 is the MD5 sum of the raw content of the file. This can be used to link a version with the on-disk file content stored in the content cache. It can also be used as a check to see whether two versions are identical.

Reporting Views

For databases that support the concept of a "View," PeerReview Complete creates a set of Views specifically for the purpose of external report-writers. You should use these Views whenever possible; we will make sure that the definitions of these Views remain the same even if we change the database schema in future versions.

You can also use these Views as a guide for how to create other custom queries.

Here are the special reporting Views:

`assignment_view`	Contains columns from the assignment table which will be maintained in the event of a future schema change.
`assignment_state`	Contains names and descriptions for the codes used in the `actioncode` column of the `assignment` table.
`defects_by_path`	One row per review reporting file path, lines of code reviewed, and number of defects.
`defectcustom`	These are the custom fields you've defined for defects, one row for each defect. If you change the custom field definition, the layout of this table will change as well (automatically, and immediately). Warning: Because the exact custom field titles are used for column names, if you change the title of a custom field it will change the definition of this view. With review workflows 112, each review might have a different subset of custom fields. In this case fields are `NULL` when they are not applicable.

© 2003-2011 SmartBear Software

`defectcustom_compat`	Same as `defectcustom`, but column names are in the form `custom_id_N` where "N" is the custom field ID as displayed in the custom fields admin page 104. The columns are ordered exactly the same as `defectcustom`, so you can also tell which is which by comparing the two views.
`defect_custom_dropdowns`	This table contains a list of the possible values as defined in drop-down and multi-select Defect Custom Fields.
`defect_state`	Contains "defect state" codes and names. Join on this table if you would like to display more user-friendly defect state names.
`defect_view`	Contains columns from the defect table which will be maintained in the event of a future schema change.
`participant_singleline_values`	Shows Participant Custom Field values that users have selected in reviews. This table only shows values for "Single-Line Text" type custom fields. A NULL value means that a custom field was defined for a review's template, but the user did not specify a value.
`participant_multiline_values`	Shows Participant Custom Field values that users have selected in reviews. This table only shows values for "Multi-Line Text" type custom fields. A NULL value means that a custom field was defined for a review's template, but the user did not specify a value.
`participant_select_values`	Shows Participant Custom Field values that users have selected in reviews. This table shows values for "Drop-Down" and "Multi-Select" type custom fields. A NULL value means that a custom field was defined for a review's template, but the user did not specify a value. In the case of Multi-Select fields, if a user selected multiple values then multiple rows will appear in the results, one for each selection.
`phase`	Represents the various phases a review can be in.
`review_activity`	For each unique combination of review, user, and role, reports the person-hours 455 spent. This includes data for current review participants only. If a user was a participant but isn't now, that person will not be included in this result. However that time will be included in the `review_activity_summary` view.
`review_activity_summary`	One row per review reporting author, reviewer, rework hours, and total person-hours 455 spent in the review. • total_person_hours -- a total of all time spent in the review • author_rework_hours -- time spent by the author during the "rework" phase. • author_hours -- time spent by the author outside of the rework phase.

© 2003-2011 SmartBear Software

	• reviewer_hours -- total time spent by reviewers. (active & passive) • active_reviewer_hours -- time spent by "active" reviewers (i.e.: reviewer required to finish a review.) • passive_reviewer_hours -- time spent by "passive" reviewers (i.e.: those not required to finish a review.)
`review_comment_summary`	One row per review reporting author, reviewer, and total number of comments made in the review.
`review_defect_summary`	One row per review reporting the number of defects created in that review.
`review_metrics_summary`	One row per review reporting a variety of standard metrics such as inspection rate, defect rate, defect density, and the individual numbers used to form those ratios. Some values will contain `NULL` values because of divide-by-zero errors.
`review_version_list`	Lists all versions actually associated with a review, although with the associated review.
`review_version_summary`	One row per review reporting the number of files and line metrics 455 (total, added, modified, removed) for all files in the review.
`reviewcustom`	These are the custom fields you've defined for reviews, one row for each review. If you change the custom field definition, the layout of this table will change as well (automatically, and immediately). With review workflows 112, each review might have a different subset of custom fields. In this case fields are `NULL` when they are not applicable.
`reviewcustom_compat`	Same as `reviewcustom`, but column names are in the form `custom_id_N` where `"N"` is the custom field ID as displayed in the custom fields admin page 104. The columns are ordered exactly the same as `reviewcustom`, so you can also tell which is which by comparing the two views.
`review_custom_dropdowns`	This table contains a list of the possible values as defined in drop-down and multi-select Review Custom Fields.
`review_view`	Contains columns from the review table which will be maintained in the event of a future schema change.
`role_view`	Contains columns from the role table which will be maintained in the event of a future schema change.
`userprefs`	Additional user preferences, one row per user. Most user information and preferences are stored in the `user` table; the rest is here.

© 2003-2011 SmartBear Software

`userprefs_compat`	Same as userprefs, but column names are in the form custom_id_N where "N" is an internal ID used to store user preferences. The columns are ordered exactly the same as userprefs, so you can tell which is which b comparing the two views.
`user_view`	Contains columns from the user table which will be maintained in the event of a future schema change.

6.1.2 Example SQL: Participant Custom Fields

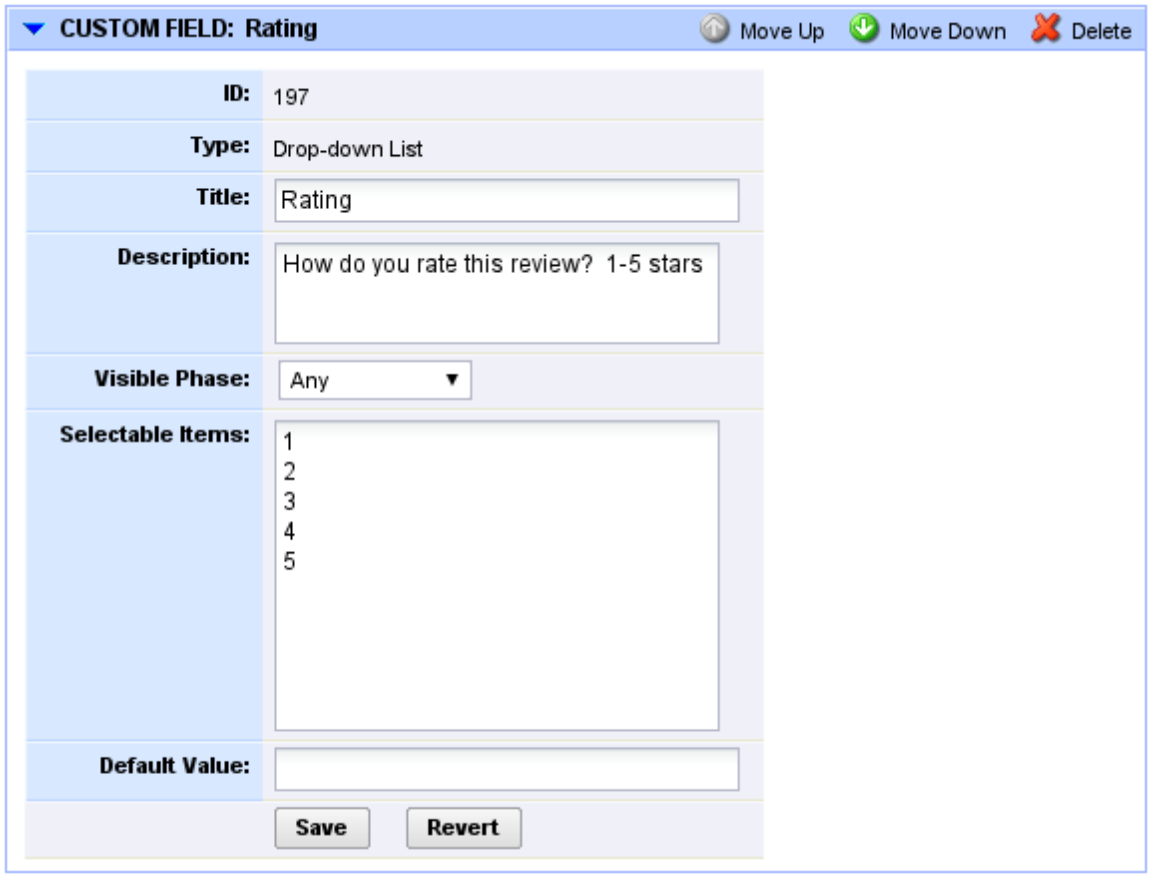
What Are Participant Custom Fields?
In PeerReview Complete, we introduced Per-ParticipantCustom Fields 104. This allows you to create a field in a review (via its review template) to which each review participant can assign his or her own value. This allows collection of any type of data that may be different for each participant.

Representing this data in table/column style is sometimes less than useful, especially since, generally, you will want to perform some calculation on the data before displaying it. To aid you in this, we have created views in the database to help you retrieve these values and perform calculations on them.

Examples in MySQL
Below is a sample of how to query Participant Custom Fields inMySQL 25, using MySQL Workbench to connect directly to the database. The syntax will be similar for Oracle and MSSQL, though you may need to reference the documentation for your particular database. You will also need to use a SQL query tool that is able to connect to those databases.

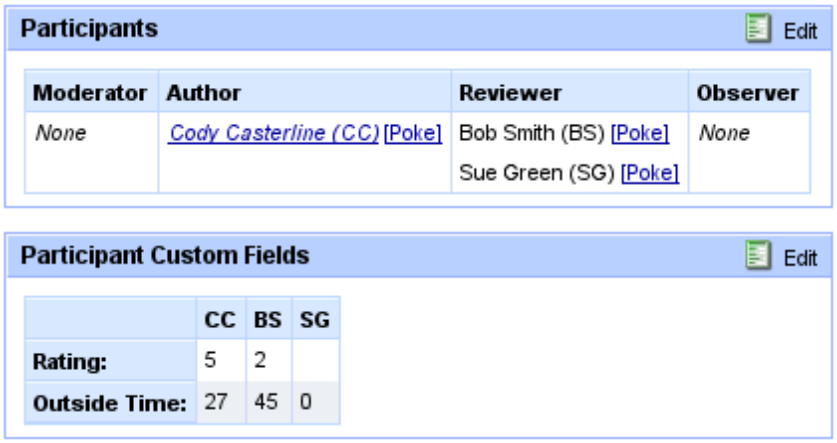
Participant Custom Fields
In these examples, we'll use the following Participant Custom Fields:

© 2003-2011 SmartBear Software



© 2003-2011 SmartBear Software

We'll also use a review (#3919) with some sample data entered for the custom fields:



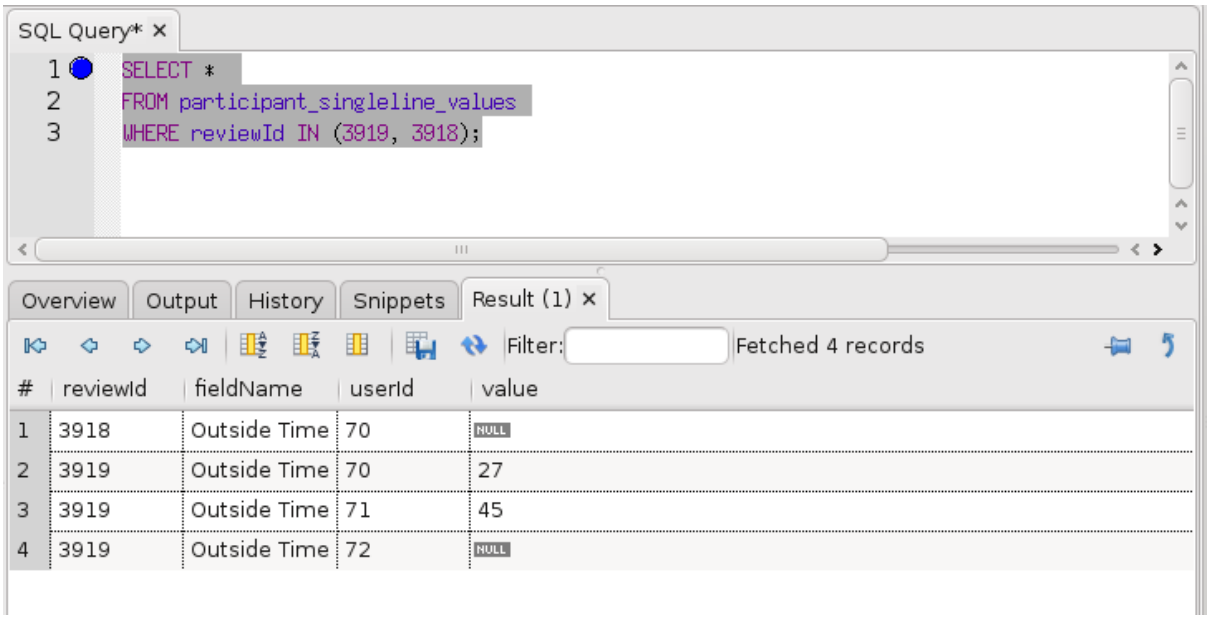
Simple Data

The simplest method of selecting the data from the database is just to show the values as they were entered by users. To do this, we'll use one of the views 427 for Participant Custom Fields that have been defined in the database.

```
SELECT  *
```

© 2003-2011 SmartBear Software

```
FROM   participant_singleline_values
WHERE  reviewId  IN  (3919,  3918);
```



Note that, for the purposes of these examples, we're limiting the results to two reviews.

Review #3918 only has one participant (it's still in planing phase), who hasn't yet specified a value for Outside Time.

Review #3919 has three users, two of which have specified values for Outside Time.

A Note About Database Data Types

PeerReview Complete stores all values for Custom Fields as strings, or sequences of letters and numbers. This is fine for the purposes of displaying the data (as above), but if you want to perform calculations on the data, you will need to tell your database to convert the strings into a meaningful number type, such as an integer, or a floating point number.

In MySQL, you perform this data conversion using the CONVERT() function.

SQL also contains a 'NULL' value, which represents the absence of a specified value. If you want to replace NULL values with some meaningful default value, use the IFNULL() function. Generally, aggregate functions like SUM() and AVG() will exclude NULL values from their calculations, so you won't need this function.

Sums

If we would like to find the sum of Outside Time for each review, we can modify the query to be:

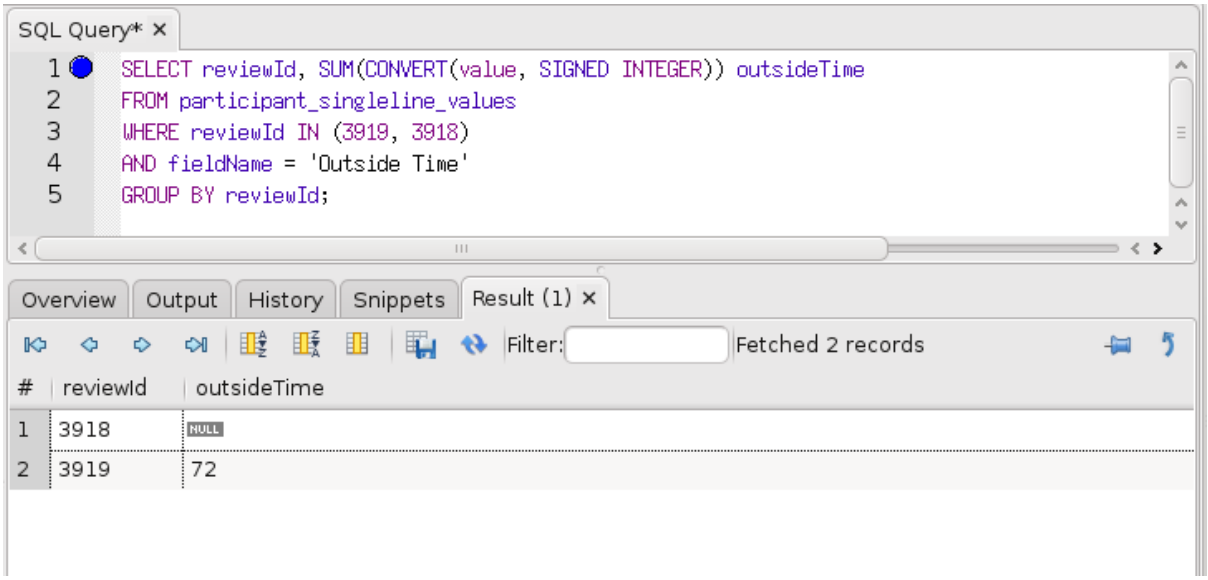
```
SELECT  reviewId,  SUM(CONVERT(value,  SIGNED  INTEGER))  outsideTime
FROM   participant_singleline_values
WHERE  reviewId  IN  (3919,  3918)
AND  fieldName  =  'Outside  Time'
GROUP  BY  reviewId;
```

© 2003-2011 SmartBear Software

In the SELECT line, we now explicitly list each of the columns we wish to select. We use the CONVERT() function to convert the values to (signed) integers, and then the SUM() function to add them up. We also specify that the summed value should be named "outsideTime".

In the WHERE clause, we've added an additional 'AND' clause. This makes sure that we only sum up values of the correct field type: 'Outside Time'.

Because we're using an aggregate function (SUM()), we have to specify which pieces of data we want to perform that calculation on. In this case, we want the sum for each review, so we specify "GROUP BY reviewId". The database server first groups all results with the same review ID together, then sums the values for each of those groups.

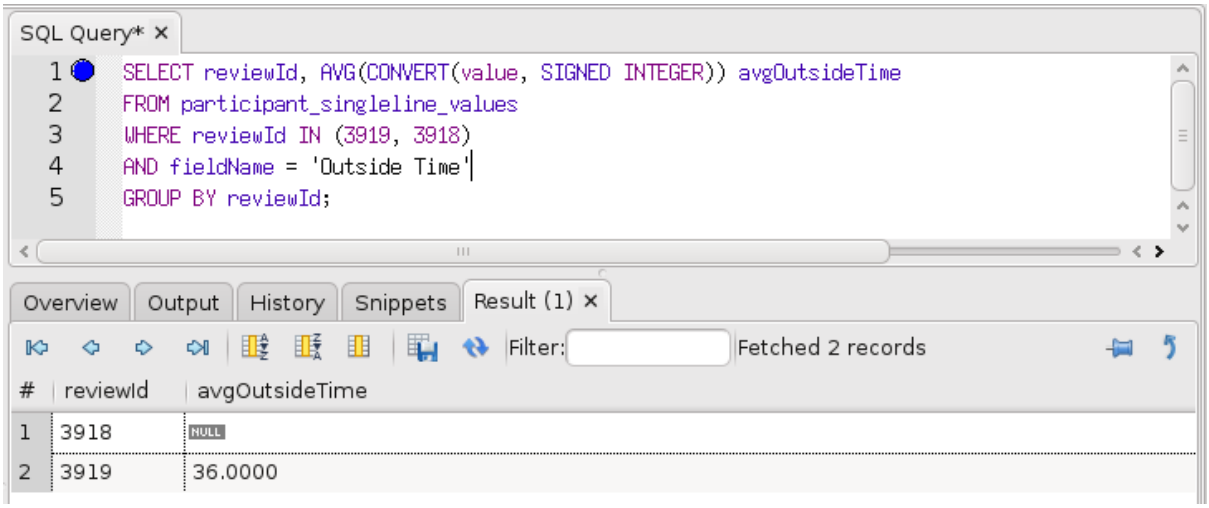


Averages

To perform averages, the query is nearly identical, we just replace the SUM() function with the average function, AVG():

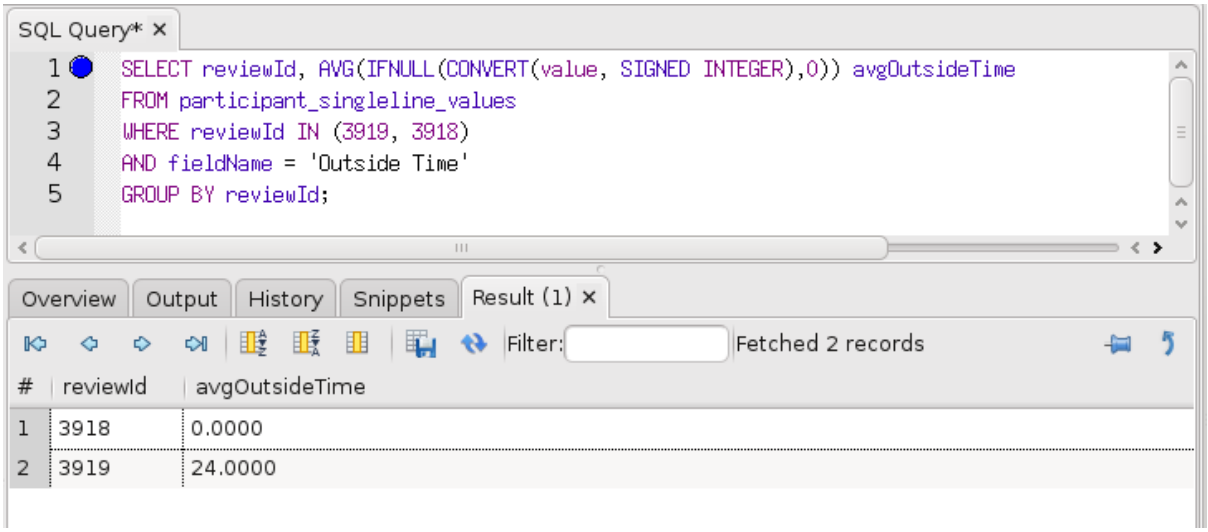
```
SELECT reviewId, AVG(CONVERT(value, SIGNED INTEGER)) avgOutsideTime
FROM participant_singleline_values
WHERE reviewId IN (3919, 3918)
AND fieldName = 'Outside Time'
GROUP BY reviewId;
```

© 2003-2011 SmartBear Software


SQL Query* ×
1 SELECT reviewId, AVG(CONVERT(value, SIGNED INTEGER)) avgOutsideTime
2 FROM participant_singleline_values
3 WHERE reviewId IN (3919, 3918)
4 AND fieldName = 'Outside Time'
5 GROUP BY reviewId;
Overview Output History Snippets Result (1) ×
Filter: Fetched 2 records
reviewId avgOutsideTime
1 3918 NULL
2 3919 36.0000

Note that these may or may not be the results you were expecting, depending on how you wish to treat NULL values. In the above results, NULL values have been excluded from the calculation of the average, so what we're really seeing is the average of 27 and 45. If you instead want to treat NULL values as if they were 0, you need to specify that in your query using the IFNULL() function. IFNULL(x, y) says that if x is NULL, return y. (Otherwise, it just returns x.) So we'll modify our query like this to treat NULLs as if they were 0:

```
SELECT  reviewId,  AVG(IFNULL(CONVERT(value,  SIGNED  INTEGER),0))  avgOutsideTime
FROM   participant_singleline_values
WHERE  reviewId  IN  (3919,  3918)
AND  fieldName  =  'Outside  Time'
GROUP  BY  reviewId;
```


SQL Query* ×
1 SELECT reviewId, AVG(IFNULL(CONVERT(value, SIGNED INTEGER),0)) avgOutsideTime
2 FROM participant_singleline_values
3 WHERE reviewId IN (3919, 3918)
4 AND fieldName = 'Outside Time'
5 GROUP BY reviewId;
Overview Output History Snippets Result (1) ×
Filter: Fetched 2 records
reviewId avgOutsideTime
1 3918 0.0000
2 3919 24.0000

Note that, now that we treat missing (NULL) values as zeros, the average Outside Time for Review #3919 is the average of 27, 45, and 0. Also note that the missing values for Review #3918 are also interpreted as zeros, so we get a 0 result instead of a NULL.

© 2003-2011 SmartBear Software

6.1.3 Example XPath and XSL

If you feel more comfortable handling XML conversions than connecting directly to a database, you can instead get data about reviews using the command ccollab admin review-xml 235. This command has one required argument, the review ID. By default, it will output the entire XML document for the review. However, there are optional arguments of --xpath and --xsl-file which allow you to query the resulting document for particular information.

XPath

XPath allows you to address XML elements in an XML document much like paths in a file system. The hierarchy in an XPath expression tells the XPath parser where in the XML document to find elements.

For example, to get the section of the review-xml document that contains participant custom fields, you would use an XPath expression like this:

```
/reviews/review/participant-custom-fields
```

Here's an example command invocation and its response:

```
$ ccollab admin review-xml 3919 --xpath /reviews/review/participant-custom-fields

<participant-custom-fields>
     <user userId="70">
           <outside-time metaDataId="198" title="Outside Time">27</outside-time>
           <affected-components metaDataId="199" title="Affected Components">APIs
Business Logic
Database  Back-end</affected-components>
          <rating metaDataId="197" title="Rating">5</rating>
     </user>
     <user userId="71">
           <outside-time metaDataId="198" title="Outside Time">45</outside-time>
           <affected-components metaDataId="199" title="Affected Components"/>
          <rating metaDataId="197" title="Rating">2</rating>
     </user>
     <user userId="72">
           <outside-time metaDataId="198" title="Outside Time">0</outside-time>
           <affected-components metaDataId="199" title="Affected Components"/>
         <rating metaDataId="197" title="Rating"/>
     </user>
</participant-custom-fields>
```

Finding Particular Values

To get all values of a particular type, you'll need to specify a few more path elements. Note in the sample above that all of the Participant Custom Fields are grouped by the user that specified them. Within that user tag, each of the fields gets a unique XML tag which is a normalized form of the title of the custom field title. The value for that field is stored as text inside of that field's tag. So, to get the values for the Rating field, you would use the following XPath expression:

```
/reviews/review/participant-custom-fields/user/rating/text()
```

Here's the example invocation, and its output:

```
ccollab admin review-xml 3919 --xpath "/reviews/review/participant-custm-fields/user/rating/text()"

5
2
```

© 2003-2011 SmartBear Software

Note that we had to quote the XPath expression because some command-line shells interpret parentheses as non-literal characters.

XSLT

XSLT (Extensible Stylesheet Language Transformation) is a subset of XSL which can be used to transform an XML document into another format. You can use XPath expressions along with XSLT to perform more complicated data queries. For example, if we want to show all "rating" values and the user ID of the user who set them, we can use a file like this:

File: sample.xslt

```
<?xml  version="1.0"  encoding="iso-8859-1"?>
<xsl:stylesheet   version="1.0"   xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output  method="text"/>
<xsl:template  match="/">
     <!-- Output CSV of Participant Custom Fields -->

     <!-- Header: -->
      <xsl:text>UserID</xsl:text>
      <xsl:text>, </xsl:text>
      <xsl:text>Field Name</xsl:text>
      <xsl:text>, </xsl:text>
      <xsl:text>Field Value</xsl:text>
      <xsl:text>&#10;</xsl:text><!-- linefeed -->

        <xsl:for-each  select="/reviews/review/participant-custom-fields/user">
           <xsl:value-of select="@userId"/>
          <xsl:text>, </xsl:text>
           <xsl:value-of select="rating/@title"/>
          <xsl:text>, </xsl:text>
           <xsl:value-of select="rating/text()"/>
           <xsl:text>&#10;</xsl:text>
     </xsl:for-each>

</xsl:template>
</xsl:stylesheet>
```

Here's how you would use the above XSL file, and some sample results:

```
$ ccollab  admin  review-xml  3919  --xsl-file  sample.xslt

UserID,  Field  Name,  Field  Value
70,  Rating,  5
71,  Rating,  2
72,  Rating,
```

Note that user #72 has not specified a value for Rating, so that value is empty.

6.2 Bug-Tracking Integration

PeerReview Complete can integrate with external issue-tracking systems such as TeamTrack, Bugzilla, JIRA, and FogBugz. Integration can be done in several ways depending on the type of integration you require.

© 2003-2011 SmartBear Software

Hyperlinked Issues
PeerReview Complete can identify external issue references in titles, custom fields, comments, defects, and more, and automatically hyperlink them to the right web page in your external issue tracker.

For example, here at SmartBear we use the word "`Case`" followed by a number to identify a issue in our bug system. So we configured PeerReview Complete to search for text in the (regular expression) form of "`\bcase\s*(\d+)`" and automatically hyperlink the number part (in the grouping symbols) to FogBugz, our external issue tracker.

See this section 117 for details on how to do this configuration.

Referring to issues in the Review Overview
It is common to want to associate one or more issues with each review. The easiest way to do this is to create a custom review field 104 and set the regular expression validator to ensure that only properly-formatted issues are entered in.

You should also set up the issue-hyperlink feature described above so that these fields are interactive.

Externalizing: Moving a defect from PeerReview Complete to an external issue tracker
Sometimes you find a defect during review that you don't want to fix just now. In this case you want to move the defect from PeerReview Complete into an external issue tracker. You also want to record this state and audit trail in the review.

The externalized defects 171 feature allows you to do just that. See that section of the manual for details.

It is unusual to integrate beyond this point
This is the extent to which most of our customers go with integration. Most people don't want review defects mirrored into an external issue tracker because these defects were never "delivered."

For example, if you had done the peer review side-by-side with someone, you wouldn't enter in defects then -- you'd just fix them!

Usually the QA department runs off the issue tracker for verification. How can QA verify something that wasn't necessarily externally broken (e.g. some function wasn't checking input values carefully) or something just in the file (e.g. some method wasn't documented properly).

Mirroring defects from PeerReview Complete into an external issue tracking system
You can use the server-side trigger 115 that runs every time a defect is added, edited, marked fixed, or deleted. Your trigger would locate the corresponding external issue record and update it. This integration takes a bit of work. We don't have it pre-built because everyone's external issue-tracking system is different.

For an example see this scripting tutorial 445.

6.3 Scripting

Scripting is the easiest way to implement custom behaviors with PeerReview Complete.

© 2003-2011 SmartBear Software

Command Line Client

You can use the Command Line Client to script actions on your user's machines, or install it on the same machine as the PeerReview Complete Server and invoke it with a trigger 115.

The Command Line Client can create 240, delete 242, and edit 241 reviews 239, including templates 241 and review custom fields 241. It can assign 245 or remove 246 participants 245 within a review, and even move reviews to the next phase 243 (e.g. from "planning" to "inspection"). It can also create comments 247 and defects 249, including defect custom fields 250, and mark defects as external 252.

You can use the Command Line Client to create 256, edit 257, enable 258 or disable 258 user 256 accounts. It can be used to set up user information like email addresses 258 and phone numbers 258. It can also be used to set up author 258 or file-based 260 subscriptions.

Batching Commands

Often when scripting you find that you need to run multiple Command Line Client commands. It's OK to simply run them one after the other in your script, but it is more efficient to group them together using the `ccollab admin batch` 236 command. The `batch` command takes as input an XML file which lists the commands to be run and optionally global options to use to run them.

It's faster to run multiple commands using the `batch` command because the Command Line Client only has to connect and authenticate with the PeerReview Complete Server once. Also the XML input format is useful if the data you are passing to the Command Line Client contains characters difficult to encode on the command-line like line feeds, quotes, or multi-byte characters.

Extracting Data

The `ccollab admin review-xml` 235 command lets you extract data from one or more reviews in your script. You can supply an XSL file 235 to format the output, or an XPATH 235 expression to select only the specific data you need. You can even use XPATH functions to perform some computation.

You can also query the server for a list of reviews using the built-in reports 176, and then use the `ccollab admin wget` 237 command to download the CSV output format 179 to be parsed by your script. Note the CSV output format of the reports does not consume a license. For example, to get a list of the reviews currently in progress run:

```
ccollab admin wget "/go?page=ReportReviewList&formSubmittedreportConfig=1&reviewIdVis=y&
```

Prompting

You can disable all interactive prompting by specifying the `non-interactive` 205 global option.

If `non-interactive` 205 is not specified, the Command Line Client will automatically prompt for server `url` 205, `user` 205, and `password` 205 as necessary when it connects to the PeerReview Complete Server. You can also use the special value "*ask*" when specifying a `review`. Using "*ask*" will cause the Command Line Client to interactively prompt the user with the list of reviews they are currently involved in.

If you are invoking more than one command, you can use the special value "*last*" in combination with "*ask*". Using "*last*" will cause the Command Line Client to address the review most recently accessed by the Command Line Client. Typically the first command in the script will use "*ask*" and then subsequent commands use "*last*" to access the same review.

© 2003-2011 SmartBear Software

Opening a Browser
PeerReview Complete is fundamentally a web-based tool and the actual reviewing goes on in a browser. Many Command Line Client commands automatically open a browser to allow the user to take the next step. You can prevent the Command Line Client from opening a browser by specifying the`no-browser` 205 global option.

You can also explicitly open a browser using the`ccollab browse` 232 command. Note the `browse` command ignores the `no-browser` 205 global option.

"Virtual" Script Users

Often when setting up scripting it is convenient to create a PeerReview Complete user which the script will use to log in to the PeerReview Complete server. Simply create a user in the normal way and give the credentials (login and password) to the script. The command-line scripting commands will not cause the "virtual" script user to consume a license (except `ccollab admin wget` 237, if the URL it's loading consumes a license).

Downloading file version content
A special URL lets you download the binary contents of any file version efficiently. The URL is authenticated, so you have to use `ccollab admin wget` 237. Note the version content URL does not consume a license. In this example, the file version ID is `12345`:

```
ccollab admin wget /data/server?versionid=12345
```

6.3.1 Mirroring Defects to an external issue-tracker

In this tutorial we will set up a script that will automatically mirror defects found during a peer review in PeerReview Complete to an external issue-tracking system. In this tutorial we will mirror defects to FogBugz, but the steps in this tutorial can be modified to mirror defects to any external issue-tracker.

Outline
When a Defect is created in PeerReview Complete the Defect activity trigger 115 invokes a script which creates a new "Case" in FogBugz with a link back to the review. Then the Defect in PeerReview Complete is marked "external" with a link to the mirrored Case in FogBugz.

Prerequisites
The script in this example is written in Perl, so the server must have the Perl runtime installed. The script invokes the PeerReview Complete Command Line Client, so that needs to be installed on your server machine as well. Be sure to install these prerequisites so that they are accessible and executable by the system user which is running the PeerReview Complete server (this is especially important on Unix systems).

Script Step 1: Set up constants
The script will need to know a few values to do it's work. These will be hard-coded in to the script:

```
# URL to the PeerReview Complete Server
$CCOLLAB_URL = "http://localhost:8080";

# Login of PeerReview Complete User who will mark the Defect as external
$CCOLLAB_USER = "admin";

# Password of PeerReview Complete User who will mark Defect as external
$CCOLLAB_PASSWORD = "";
```

© 2003-2011 SmartBear Software

```
# URL to the FogBugz server
$FOGBUGZ_URL = "http://bugs";

# Email of user to use to create Case in to FogBugz
$FOGBUGZ_USER_EMAIL = "person\@yourcompany.com";

# Password of user to use to create Case in to FogBugz
$FOGBUGZ_USER_PASSWORD = "yourpasswordhere";
```

Script Step 2: Read parameters from command-line

The Review ID, Defect ID, and Defect title will be supplied by PeerReview Complete as parameters when it invokes the script from the trigger (we set that up in Step 9 447). The script needs to read those parameters from the command-line:

```
# read parameters from command-line
$reviewId = $ARGV[0];
$defectId = $ARGV[1];
$defectTitle = $ARGV[2];
```

Script Step 3: Logon to FogBugz

The script will use the FogBugz web services API to mirror the Defect in to FogBugz. In order to use the API we first have to "logon" and acquire an "authentication token":

```
# logon to FogBugz
my $response = get("$FOGBUGZ_URL/api.php?cmd=logon&email=$FOGBUGZ_USER_EMAIL&password=$FOGBUGZ_USER_P

# extract authentication token
$response =~ /<token>(?:<!\[CDATA\[)?([^\]]+)(?:\]\]>)?<\/token>/;
my $token = $1;
```

Script Step 4: Create Case in FogBugz

The script creates a "Case" in FogBugz, setting the title of the Case to the text from the PeerReview Complete Defect, and putting a link back to the PeerReview Complete server in the description of the Case. Note that the Defect title and Case description have to be escaped because they are going in to a "get" URI. The script extracts the resulting Case ID and saves it for later:

```
# create new Case
$defectTitle = uri_escape($defectTitle);
$description = uri_escape("Defect D$defectId mirrored from PeerReview Complete Review $reviewId ($
$response = get("$FOGBUGZ_URL/api.php?token=$token&cmd=new&sTitle=$defectTitle&sEvent=$description");

# extract Case ID
$response =~ /ixBug="(\d+)"/;
my $case = $1;
```

Script Step 5: Logoff from FogBugz

The script logs off from FogBugz, invalidating the authentication token:

```
# log off from FogBugz (invalidate authentication token)
$response = get("$FOGBUGZ_URL/api.php?token=$token&cmd=logoff");
```

© 2003-2011 SmartBear Software

Script Step 6: Mark PeerReview Complete Defect as "external"
The script runs the Command Line Client `ccollab admin review defect mark-external` 252 command to mark the Defect as external. The "external-name" of the Defect is the Case number in FogBugz (which will appear as a link after Step 8 447):

```
# mark Defect as "external" in PeerReview Complete
$ccollabOptions = "--url $CCOLLAB_URL";
$ccollabOptions .= " --user $CCOLLAB_USER";
$ccollabOptions .= " --password \"$CCOLLAB_PASSWORD\"";
$ccollabOptions .= " --quiet --non-interactive";
system("ccollab $CCOLLAB_OPTIONS admin review defect mark-external $defectId \"Case $case\"");
```

This step in the script works fine if you have configured PeerReview Complete to work with MySQL. With any other database, the process is slightly more complicated - please contact support@smartbear.com for more details.

Step 7: Test the script from the command-line
That's our whole script! Here it is in finished form: `mirror-defect.pl`. Don't forget to replace the constants in the script (urls, users, passwords etc...) with the appropriate values for your environment, then copy the script to an accessible place on your server. The script needs to be readable by the system user which is running the PeerReview Complete server.

Before you configure the PeerReview Complete server to invoke the script automatically, test it manually by opening a console on your server machine running it on the command-line. **Be sure to log in to your server machine as the user which is running the PeerReview Complete server.** First create a review and create a defect in the review, then run the script.

For example, if you created Review 1234 with Defect D5678, then run the script with this command:

```
C:\Perl\perl.exe "C:\Program Files\PeerReview Complete Server\mirror-defect.pl" 1234 5678 "mirrored De
```

You should see output similar to this:

```
Mirrored Defect D5678 in FogBugz as Case 91011
Connecting to PeerReview Complete Server http://localhost:8080
Connected as: Collaborator Administrator (admin).
D5678 marked as external
```

When you refresh your browser in PeerReview Complete you should see the Defect has been marked as external, with the description "Case 91011".

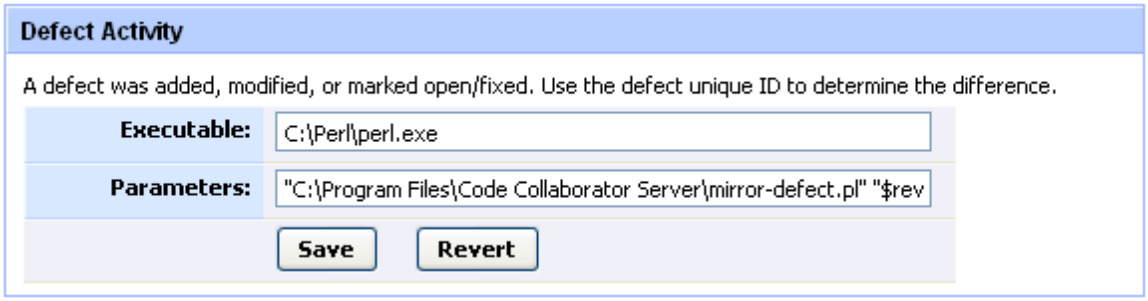
Step 8: Set up Bug-Tracking integration link
Using the PeerReview Complete built-in Bug-Tracking hyperlink function, set up the text "Case <number>" to automatically hyperlink to that Case number in FogBugz 443. Now if you go back to look at Defect D5678 in PeerReview Complete the text "Case 91011" should be a hyperlink to Case 91011 in FogBugz.

Step 9: Invoke Script from a trigger
Make sure you have Step 7 447 working before you move on to this step. The last thing we need to do is tell the PeerReview Complete server to automatically invoke the `mirror-defect.pl` script when a Defect is created. We do this with the Defect Activity trigger 115. We will use these values:

Executable: `C:\Perl\perl.exe`

© 2003-2011 SmartBear Software

Parameters: `"C:\Program Files\merge2.2\specs\collab\Baggage\mirror-defect.pl" "$review.id" "$defect.id" "$defect.txt"`



Defect Activity Trigger

Note that it is important to use the FULL PATH of both the Perl runtime and the `mirror-defect.pl` script. Also note that the parameters are enclosed in quotes in case they have spaces in them. The Review id 119, Defect id 121, and Defect text 121 come from substitution variables 119.

Finished

That's it! Now when anyone creates a Defect in PeerReview Complete it will be automatically mirrored to FogBugz, and the Defect will be "marked as external" with a link to the Case in FogBugz. Note that for performance reasons the trigger runs *after* the Defect is created in a separate thread, so you may have to refresh your browser a couple of seconds after creating the Defect to see it automatically externalized.

6.3.2 Syncing users from Perforce

In this tutorial we will write a script that will create a corresponding PeerReview Complete user for every Perforce user. The script can be run periodically to pick up any new Perforce users.

Outline

The script runs the Perforce `p4 users` command, parses the output, and then calls the PeerReview Complete Command Line Client `ccollab admin user create` 256 command for every user. If the user already exists then the `ccollab admin user create` command fails and the script goes on to the next user.

Prerequisites

The script in this example is written in Perl, so you must have the Perl runtime installed. The script invokes the PeerReview Complete Command Line Client, so that needs to be installed as well. Note you must be a PeerReview Complete Administrator in order for this script to work.

Script Step 1: Get users from Perforce

The script runs the `p4 users` command to get the a list of all user records from Perforce.

```
# Get users from Perforce
@p4Users = `p4 users`;
```

Script Step 2: Parse fields from Perforce user record

The script extracts the user name, email, and full name from the Perforce user record.

```
#parse fields from Perforce user record
$p4User  =~  /(\S+)\s*<([^>]*)>\s*\((.*?)\)\s*accessed.*/;
```

© 2003-2011 SmartBear Software

```
$user = $1;
$email = $2;
$fullName = $3;
```

Script Step 3: Create user in PeerReview Complete

The script runs the Command Line Client`ccollab admin user create` 256 command to create the user in PeerReview Complete. If a user with that name already exists, the command fails and the script goes on to the next user.

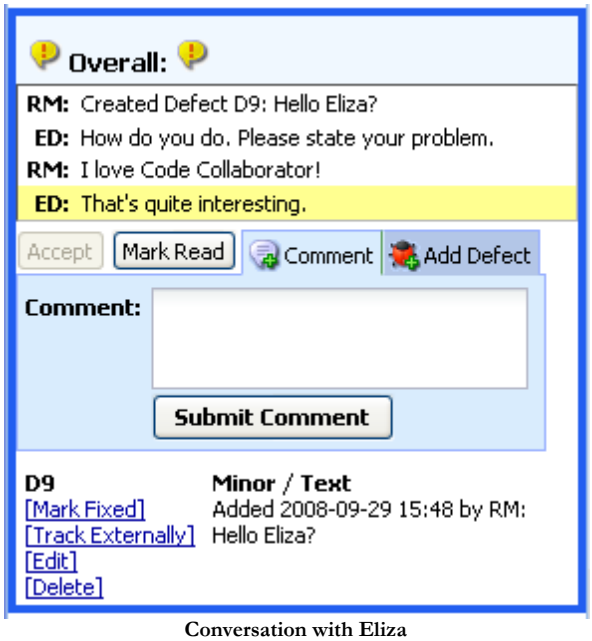
```
#create user in PeerReview Complete - does nothing if user already exists
system("ccollab admin user create \"$user\" --email \"$email\" --full-name \"$fullName\"");
```

Finished

That's it! Here's the script in finished formsyncusers-p4.pl. You can run this script periodically (e.g. with a `cron` job) and it will keep your Perforce users and PeerReview Complete in sync.

6.3.3 Conversation with Eliza

In this tutorial we will set up a script that will let a user have a conversation with the classic ELIZA program. Eliza will appear to chat using comments in a Review. Eliza will only respond to comments in conversations that have a Defect containing her name.



Conversation with Eliza

Outline

When a Defect is created in PeerReview Complete theDefect Activity trigger 115 launches a Perl script in the background. The script polls PeerReview Complete for comments using the Command Line Client. When it finds a new comment, it passes the comment to an implementation of the ELIZA program, and then posts Eliza's response to the conversation using the Command Line Client.

The script runs indefinitely until the Defect is marked fixed or deleted, or the Review is finished.

© 2003-2011 SmartBear Software

Prerequisites

The script in this example is written in Perl, so you must have the Perl runtime installed.

The Perl script invokes the PeerReview Complete Command Line Client, so that needs to be installed on your server machine. Be sure to install the Command Line Client so that it is accessible and executable by the system user which is running the PeerReview Complete server (this is especially important on Unix systems).

The script uses a clone of ELIZA written in Perl called Chatbot::Eliza which must be installed.

Script Step 1: Set up constants

The script will need to know a few values to do it's work. These will be hard-coded in to the script:

```
# URL to the PeerReview Complete Server
$CCOLLAB_URL  =  "http://localhost:8080";

# Login of PeerReview Complete User who Eliza will use for chat
# should be a PeerReview Complete administrator
$CCOLLAB_USER  =  "eliza";

# Password of PeerReview Complete User who Eliza will use for chat
$CCOLLAB_PASSWORD  =  "eliza";

# Seconds to sleep before polling for new chat
$REFRESH_DELAY_SECONDS  =  4;
```

Script Step 2: Read parameters from command-line

The Review ID and Defect ID will be supplied by PeerReview Complete as parameters when it invokes the script from the trigger (we set that up in Step 9 454). The script needs to read those parameters from the command-line:

```
# read parameters from command-line
$reviewId  =  $ARGV[0];
$defectId  =  $ARGV[1];
```

Script Step 3: Create XSL Transform

The script needs to extract multiple pieces of information from the PeerReview Complete server. To do that more efficiently it uses an XSL Transform to extract everything in a single call to ccollab admin review-xml 235 (Step 6 451). The XSL is inlined in the Perl script using "here-document" syntax.

```
#xslt file we will use to extract info from PeerReview Complete
$xslt  =  <<XSLT;
<?xml  version="1.0"  encoding="UTF-8"?>
<xsl:stylesheet    xmlns:xsl='http://www.w3.org/1999/XSL/Transform'    version='1.0'>

       <xsl:template    match='//reviews/review'>

              <!--   get review phase -->
              <xsl:value-of    select='general/phase'/><xsl:text>
</xsl:text>

              <!--   find defect -->
              <xsl:for-each     select='defects/defect[\@defectId=$defectId]'>

                     <!--   get defect status -->
                     <xsl:value-of    select='status'/><xsl:text>
```

© 2003-2011 SmartBear Software

```
</xsl:text>

                    <!--   get defect text-->
                    <xsl:value-of    select='text'/><xsl:text>
</xsl:text>

             </xsl:for-each>

             <!--   find the conversation we're talking on (the one with the defect) -->
             <xsl:for-each       select='conversations/conversation[defects/conversation-defect/\@defect-i

                    <!-- find last comment in conversation -->
                    <xsl:for-each    select='comments/comment[last()]'>

                           <!--   get author -->
                           <xsl:value-of    select='\@creator'/><xsl:text>
</xsl:text>

                           <!--   get text -->
                           <xsl:value-of    select='text()'/><xsl:text>
</xsl:text>

                    </xsl:for-each>

                    <!-- get file-path (empty for overall comment) -->
                    <xsl:value-of    select='\@file-path'/><xsl:text>
</xsl:text>

                    <!-- get line-number (empty for overall comment or file overall comment) -->
                    <xsl:value-of    select='\@line-number'/><xsl:text>
</xsl:text>

             </xsl:for-each>

      </xsl:template>
</xsl:stylesheet>
XSLT
```

Script Step 4: Initialize Eliza

The Chatbot::Eliza program is initialized by a simple constructor.

```
# initialize  Eliza
$eliza  =  new  Chatbot::Eliza;
```

Script Step 5: Sleep for a few seconds

The rest of the script loops until the script decides to exit (Step 7 452). Every time it starts the loop, the script first sleeps for a few seconds to make sure it doesn't put too much load on the server.

```
# sleep so we don't hammer the server
sleep($REFRESH_DELAY_SECONDS);
```

Script Step 6: Query PeerReview Complete server for info

The script invokes the `ccollab admin review-xml` 235 command, passing in the XSL created in Step 3 450 on STDIN. The output is then parsed to get the Review Phase, Defect status, Defect text, comment Author, comment text, and optionally file path and line number.

```
# Query  PeerReview  Complete  server  for  info
$pid  =  open2(*CCOLLAB_OUTPUT,  *XSL_INPUT,  "ccollab  $ccollabOptions  admin  review-xml  $reviewId  --xsl-fi
print  XSL_INPUT  $xslt;
```

© 2003-2011 SmartBear Software

```
close(XSL_INPUT);
chomp(($spacer, $reviewPhase, $defectStatus, $defectText, $author, $inputChat, $filePath, $lineNumber)
close(CCOLLAB_OUTPUT);
waitpid($pid,0);

#cleanup filePath and lineNumber
chop($filePath);
chop($lineNumber);

#debug
print "reviewPhase = $reviewPhase\n";
print "defectStatus = $defectStatus\n";
print "defectText = $defectText\n";
print "author = $author\n";
print "inputChat = $inputChat\n";
print "filePath = \"$filePath\"\n";
print "lineNumber = \"$lineNumber\"\n";
print "\n";
```

Script Step 7: Stop script if appropriate

The script runs in an infinite loop, but we don't really want it to go forever! The script exits if any of the following conditions are true:

- The Review finishes (no longer in inspection phase)
- The Defect is fixed or deleted (no longer open)
- The Defect text doesn't mention Eliza

```
# safety - quit if review isn't in Inspection phase
die ("Review is no longer in Inspection phase") if ($reviewPhase !~ /Inspection/);

# safety - quit if defect isn't open
die ("Defect $defectId is no longer open") if ($defectStatus !~ /open/);

# safety - quit if defect text doesn't mention "Eliza"
die ("Defect $defectId text doesn't mention Eliza") if ($defectText !~ /Eliza/);
```

Script Step 8: Ignore comment if appropriate

The script checks whether the comment is a "system" comment like "** Marked Read**" or if the last comment in the conversation was made by Eliza herself. In both of these cases the script skips back to the top of the loop.

```
# Eliza shouldn't respond to system messages, like "** Marked Read **" or "** Accepted **"
next if ($inputChat =~ /^\*\*/);

# Eliza shouldn't talk to herself
next if ($author =~ /$CCOLLAB_USER/);
```

Script Step 9: Get response from ELIZA

The script passes the user's comment to the ELIZA program and gets back her response.

```
# Get response from Eliza
$outputChat = $eliza->transform($inputChat);

#debug
print "$outputChat\n";
print "\n";
```

© 2003-2011 SmartBear Software

Script Step 10: Upload Eliza's response to PeerReview Complete

The script uploads Eliza's response to the conversation using the ccollab admin review comment create 247 command.

```
# build command to upload Eliza's comment to PeerReview Complete
$uploadCommand = "ccollab $ccollabOptions admin review comment create $reviewId \"$outputChat\"";

# file path is optional (no file path for overall review chat)
if ($filePath) {
        $uploadCommand .= " --file \"$filePath\"";
}

# line number is optional (no line number for overall review chat or overall file chat)
if ($lineNumber) {
        $uploadCommand .= " --line-number $lineNumber";
}

#debug
print "Running $uploadCommand\n";

# upload Eliza's comment to PeerReview Complete
system("$uploadCommand");
```

Step 11: Test the script from the command-line

That's our whole script! Here it is in finished form: `eliza.pl`. Don't forget to replace the constants in the script (url, user, password) with the appropriate values for your environment, then copy the script to an accessible place on your server. The script needs to be readable by the system user which is running the PeerReview Complete server.

Before you configure the PeerReview Complete server to invoke the script automatically, test it manually by opening a console on your server machine running it on the command-line.**Be sure to log in to your server machine as the user which is running the PeerReview Complete server.** First create a Review and create a Defect in the Review with the word "Eliza" in it, then run the script.

For example, if you created Review 1234 with Defect D5678, then run the script with this command:

```
/usr/bin/perl /home/rpaterson/eliza.pl 1234 5678
```

You should see output similar to this:

```
reviewPhase =    Inspection
defectStatus = open
defectText = Hello Eliza?
author = rpaterson
inputChat = Created Defect D11: Hello Eliza?
filePath = "SymlinkTest.java"
lineNumber = "73"

How do you do. Please state your problem.

Running ccollab --url http://localhost:8080 --user eliza --password "eliza"
 --quiet --non-interactive admin review comment create 1234
"How do you do. Please state your problem." --file "SymlinkTest.java" --line-number 73
```

The script will loop until you mark the Defect fixed or deleted, or finish the Review. When you refresh your browser in PeerReview Complete you should see Eliza's comment.

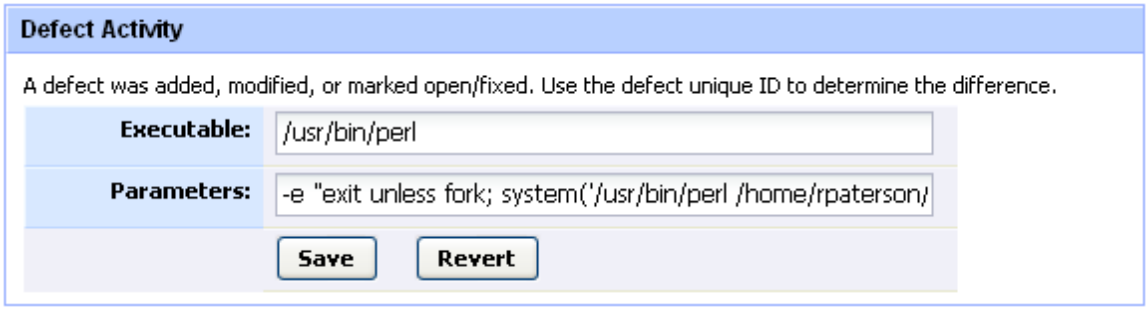
© 2003-2011 SmartBear Software

Step 12: Invoke Script from a trigger

Make sure you have Step 11 453 working before you move on to this step. The last thing we need to do is tell the PeerReview Complete server to automatically invoke the `eliza.pl` script when a Defect is created. We do this with the Defect Activity trigger 115. We will use these values:

Executable: `/usr/bin/perl`

Parameters: `-e "exit unless fork; system('/usr/bin/perl /home/rpaterson/eliza.pl ${review.id} ${defect.id} ');""`



Defect Activity Trigger

Note that it is important to use the FULL PATH of both the Perl runtime and the `eliza.pl` script. Also note the quotes and spaces - they're important. The Review id 119, and Defect id 121 come from substitution variables 119. The Perl snippet included in the Parameters field causes the script to be launched in a background process.

Finished

That's it! Now when anyone creates a Defect in PeerReview Complete the script will be invoked. If the Defect has the word "Eliza" in it then the script will monitor that conversation for comments and have Eliza respond to them.

7 Techniques & Best Practices

At SmartBear, we're experts in all kinds of peer review.

In this part of the manual, we cover techniques and best practices in code review and in PeerReview Complete.

Topics covered include:

Metrics: Definitions 455
Which metrics should you collect during reviews? Which metrics are collected automatically by PeerReview Complete? How are they calculated and what exactly do they mean?

Metrics: Analysis 456
What do you do with raw metrics numbers? How do you collect those which aren't collected automatically? What can metrics really tell us? Where might they lead us astray?

Defect Management 442
When should defects be kept in PeerReview Complete and when should they go elsewhere? What are the tools and best practices for this?

Tips and Tricks 458

© 2003-2011 SmartBear Software

Are there any short-cuts to help me get the results I want?

7.1 Metrics: Definitions

PeerReview Complete collects a variety of raw metrics automatically. This section defines these metrics; a later section 456 discusses what these metrics can tell us.

Lines of Code
The most obvious raw metric is "number of lines of source code." This is "lines" in a text-file context. Often this is abbreviated "LOC."

PeerReview Complete does not distinguish between different kinds of lines. For example, it does not separately track source lines versus comment lines versus whitespace lines.

For code review metrics, often you usually *want* to use general lines of code and not break it down by type. Often the code comments are just as much a part of the review as the code itself -- check for consistency and ensuring that other developers will be able to understand what is happening and why.

Time in Review
How much time (person-hours) did each person spend doing the review? PeerReview Complete computes this automatically. This raw metric is useful in several other contexts, usually when compared to the amount of file content reviewed.

Developers (rightly) hate using stopwatches to track their activity, but how can PeerReview Complete -- a web server -- automatically compute this number properly?

Our technique for accurately computing person-hours came from an empirical study we did at a mid-sized customer site. The goal was to create a heuristic for predicting on-task person-hours from detailed web logs alone.

We gave all review authors and reviewers physical stop-watches and had them carefully time their use of the tool. Start the stopwatch when they began a review, pause if they break for any reason -- email, bathroom, instant messenger. The times were recorded with each review and brought together in a spreadsheet.

At the same time, we collected detailed logs of web server activity. Who accessed which pages, when, etc.. Log data could easily be correlated with reviews and people so we could "line up" this amalgamation of server data with the empirical stopwatch times.

Then we sat down to see if we could make a heuristic. We determined two interesting things:

First, a formula did appear. It goes along these lines: If a person hits a web page, then 7 seconds later hits another page, it's clear that the person was on-task on the review for the whole 7 seconds. If a person hits a web page, then 4 hours later hits another page, it's clear that the person wasn't doing the review for the vast majority of that time. By playing with various threshold values for timings, we created a formula that worked very well -- error on the order of 15%.

Second, it turns out that humans are *awful* at collecting timing metrics. The stopwatch numbers were all over the map. People constantly forgot to start them and to stop them. Then they would make up numbers that "felt right," but it was clear upon close inspection that their guesses were wrong. Some people intentionally submitted different numbers, thinking this would make them look good (i.e. "Look how fast I am at reviewing!").

© 2003-2011 SmartBear Software

So the bottom line is: Our automated technique is not only accurate, *it's more accurate than actually having reviewers use stopwatches.* The intrinsic error of the prediction heuristic is less than the error humans introduce when asked to do this themselves.

Defect Count
How many defects did we find during this review? Because reviewers explicitly create defects during reviews, it's easy for the server to maintain a count of how many defects were found.

Furthermore, the system administrator can establish any number of custom fields 104 for each defect, usually in the form of a drop-down list. This can be used to subdivide defects by severity, type, phase-injected, and so on.

File Count
How many files did we review? Usually the LOC metric 455 is a better measure of "how much did we review," but sometimes having both LOC and number of files is helpful together.

For example, a review of 100 files, each with a one-line change, is quite different from a review of one file with 100 lines changed. In the former case, this might be a relatively simple refactoring; with tool support, this might require only a brief scan by a human. In the latter case, several methods might have been added or rewritten; this would require much more attention from a reviewer.

7.2 Metrics: Analysis

It's fine to collect metrics, but what do they tell us? It's tempting to apply them in many different contexts, but when are metrics telling us something and when are we reading too much into the numbers?

Defect Density
Defect Density is computed by: (number of defects) / (1000 lines of code).

This is the number of defects found, normalized to a unit amount of code. 1000 lines of code, or "kLOC" is often used as a standard base measure. The higher the defect density, the more defects you are uncovering.

It is impossible to give an "expected" value for defect density. Mature, stable code might have defect densities as low as 5 defects/kLOC; new code written by junior developers may have 100-200.

What can defect density tell us?

Let's make an experiment. We take a reviewer and have him inspect many different source files. Source files vary in size from 50 lines to 2000 lines. The reviewer inspects about 200 lines at a time so as not to get tired. We'll record the number of defects found for each file.

What would we expect to find? First, longer files ought to have more defects than shorter ones, simply because there's more code. More code means more that could go wrong. Second, some files should contain more defects than others because they are "risky" -- maybe because they are complex, or because their routines are difficult to unit-test, or because their routines are reused by most of the system and therefore must be very accurately specified and implemented.

© 2003-2011 SmartBear Software

If we measure defect density here, we handle the first effect by normalizing "number of defects" to the amount of code under review, so now we can sensibly compare small and large files. So the remaining variation in defect density might have a lot to do with the file's "risk" in the system. This is, in fact, the effect we find from experiments in the field.

So defect density can, among other things, determine which files are risky, which in turn might help you plan how much code review, design work, testing, and time to allocate when modifying one of those files.

Now let's make another experiment. We'll take a chunk of code with 5 known algorithm bugs and give it to various reviewers. We'll see how many of the defects each review can find in 20 minutes. The more defects a reviewer finds, the more effective that reviewer was at finding the defects. This is a simple way to see how effective each reviewer is at reviewing that kind of code.

Of course in real life the nature of the code and the amount of code under review varies greatly, so you can't just look at the number of defects found in each review -- you naturally expect more defects from a 200-line change than from a 2-line change. Defect density provides this normalization so you can compare reviewers across many reviews.

If you're comparing defect density across many reviews done by a single person, you are measuring the relative "risk" of various files and modules.

Inspection Rate
Inspection Rate is computed by: (lines of code reviewed) / (hours to do the review).

This is a measure of how fast we review code. A sensible rate for complex code might be 100 LOC/hour; generally good reviews will be in the range of 200-500 LOC/hour. Anything 800 LOC/hour or higher indicates the reviewer hasn't really looked at the code -- we have found by experiment that this is too fast to actually read and critique source code.

Some managers insist that their developers try to increase their inspection rate. After all this means "review efficiency" is improving. *This is a fallacy.* In fact, the *slower* the review is, the *better* job the reviewers are doing. Careful work means taking your time.

Instead, use inspection rate to help you predict the amount of time needed to complete some code change. If you know this is roughly a "1000-line change" and your typical inspection rate is 200 LOC/hour, you can budget 5 hours for the code review step in your development.

If anything, a manager might insist on a slower inspection rate, especially on a stable branch, core module, or close to product release when everyone wants to be more careful about what changes in the code.

Defect Rate
Defect Rate is computed by: (number of defects) / (hours to do the review).

This is the speed at which reviewers uncover defects in code. Typical values range between 5 and 20 defects/hour, possibly less for mature code, but not usually much greater.

The same caveats about encouraging faster or slower inspection rates apply also to defect rates. Read that section above for details.

© 2003-2011 SmartBear Software

Metrics Applied
If we've learned one thing about metrics and code review it is: *Every group is different, but most groups are self-consistent.* This means that metrics and trends that apply to one group don't necessarily apply to another, but within a single group metrics are usually fairly consistent.

This between-group difference can be attributed to the myriad of variables that enter into software development: The background, experience, and domain knowledge of the authors and reviewers, programming languages and libraries, development patterns at different stages of a product's life-cycle, project management techniques, local culture, the number of developers on the team, whether the team members are physically together or separate, etc..

7.3 Multiple Change Changelists

A changelist is a generic SCM concept representing a set of changes in version control. Changelists for some SCM systems like Perforce and Subversion are atomic entities, representing a single unit of changes that occur at the same time. Other SCM systems have changelists that accumulate changes to file versions over time, allowing multiple changes (versions) of any given file within the same changelist. Some examples of this latter type of changelist are ClearCase Activities, MKS Change Packages, and CMVC Tracks.

For these multiple-version changelists, the changelist represents an accumulation of changes to each of the source files in it. In most cases, in the context of a review or in thinking about what the changelist represents, users are interested in the difference represented by the accumulation of changes in the changelist, i.e., for each source file the difference between the latest version occurring in the changelist and the version content that existed before the first change in the changelist. PeerReview Complete calculates differences based on this, when a multiple-version changelist is uploaded to a review.

This can be confusing in some circumstances. If an added file is part of the changelist, and there are subsequent changes to that file in the changelist, then uploading this changelist for review will result in the latest change to that file appearing as an add in the changelist. In other words, it will have no predecessor. In an SCM system where added files are always version 1.1, a review of a changelist having version 1.5 of a file with no predecessor is incongruent with the versioning of the SCM system. Yet this is exactly what the accumulation of changes to that file in the changelist represent - a sum total of changes that did not exist before the changelist.

While it might be possible to find and upload all versions of each file that occur in a multiple-version changelist and make them available for comparison in PeerReview Complete, in our experience this makes for an unnecessarily complicated review - this is less of a peer review feature and more of a version history browser feature. If a review is to be conducted on successive revisions to a file in this way, the review should be conducted at each iteration of the changelist. Uploading the same changelist to the same review as each successive set of changes is made as part of the rework step of the review will result in all of the versions of each file being available in the it, and each version being available for inspection by the other review participants.

7.4 Tips and Tricks

This section will describe workarounds and tricks we use for PeerReview Complete.

© 2003-2011 SmartBear Software

Pool of Users
Using a pool of users allows a review to be sent out to multiple people, but only require one to finish the review. People can then "volunteer" to review the review. This is especially useful when a review needs to be finished, but it doesn't call for a specific reviewer.

While we don't have a specific feature in PeerReview Complete to do this, there's a good way to approximate review pools. To do so, set up a "pool" user whose email address is a mailing list of all the possible reviewers. This "user" is the one you should assign to the review. When the review starts, everyone gets the email. Whoever wants to "take" the review (and it can be more than one) will go into the review, click "edit" on the participant list, and change out the pool user for their own username. Then they can do the review, and when they're finished, the review is done.

This is best because you have the log of who it was that did the review. More than one CAN do it if they wish, which is a nice option.

There are no licensing problems because no one will actually log in as the pool user. If you don't log in, you don't count against licensing (in either fixed or floating), so although there's this user in the user list it never counts towards licensing!

The proposed "review pool" feature would make this "pool" an explicit concept in PeerReview Complete, so you don't need to set up a separate mailing list. Also, we could have a "take it" button on the review to make it that much easier. But the workaround above isn't too much work.

Picking reviews through the Command Line Client
When sending files to a review through the Command Line Client, use "last" instead of the review ID to pick the last review or "ask" to be prompted with choices to pick.

Custom Fields with Date and Times
PeerReview Complete does not have date or time custom fields, but you can closely approximate a date or time by using regular expression validation of single line text fields. For dates, the following regular expression requires a date in the 20th or 21st century that is approximately valid (yes, it accepts 31 days each month):

(?:19|20)\d\d-(?:0[1-9]|1[012])-(?:0[1-9]|[12][0-9]|3[01])

The following regular expression validates a time on a 24 hour clock:

(?:[01][0-9]|2[0-3]):[0-5][0-9]

The two could be combined to accept a date and time field. Be sure you set the description of the field to describe exactly the format you're looking for so that users do not have to parse the regular expression to know what to enter.

Checklists
In PeerReview Complete, checklists are not built into the system, but there are two ways people do this:

1. Have a "checklist URL" custom field in the review summary. The review creator uses this to link in checklists from an intranet. The reviewers can click that link to pull up the checklist and either print it out or place it on their screen(s) in a visible location.

2. Upload the checklist(s) to the review as regular documents. The reviewers similarly can open those files on-screen and/or print them out.

© 2003-2011 SmartBear Software

8 Appendices

Appendices:

Appendix A: Known Issues & Errata 460
Current known issues in the server and various client components, including integrations with other systems.

Appendix B: Version History 462
Complete version history for each public release of the various components.

Appendix C: Java VM Options 525
Overview of Java options useful for PeerReview Complete configuration.

8.1 Appendix A: Known Issues

These are the major known issues currently known for all PeerReview Complete components.

Many of the issues have workarounds; those are given here as well.

Get notified automatically when a new version is available!
SmartBear announces new publicly-available versions of PeerReview Complete using a mailing list.

If you want to **sign up to receive e-mails**, you can do so at the SmartBear Software Subscription Center:

http://www2.smartbear.com/SubscriptionCenter.html

Known Issues with the Server Component 24

- Case 34564: Reports have missing data. When the server is running on Linux systems, there can be issues with unavailable fonts or available fonts with sufficiently different font metrics that can cause the reporting engine to not render content.Workaround: Install the Microsoft True Type fonts (msttcorefonts on Debian-based systems).

Known Issues with the Web Browser Client 124

- Case 26177: Chat log 168 incorrectly encodes leading spaces, so sometimes when entering e.g. code snippets the text doesn't come out as expected.
  Workaround: Put a little character (e.g. a period) at the beginning of the line.
- Case 43101: When clicking on the link of a file in the Review Overview page the file will not be displayed at front if IE is set up to open pop-ups in tabs.
  Workaround: Go to Tools -> Internet Options -> Tab Settings -> Select "Always open pop-ups in a new window.
- Sometimes, after an upgrade, when a user logs in she sees a CSS page dump in the browser instead of going to the home page. Workaround: Just go directly to the homepage. Because you are already authenticated, that will work.
- Case 30131: In some very special cases the intra-line difference-engine doesn't properly highlight the inside of the line. The entire line is still highlighted, so users are still properly shown the main lines of the differences.

© 2003-2011 SmartBear Software

- Case 27502: Increasing the default font size in Firefox v1.5.0.7 or IE 7 can result in a very distracting (but still functional) user interface.
  Workaround: Change back to the default font size.
- Safari 3 is supported, but there are known issues with Safari 2.x.
- When running Firefox with the Firebug or Greasemonkey plug-in, the Side-by-Side view is sluggish to load. This is even true when Firebug is marked "disabled."
- Search results may not return all results without noting the truncation. This can happen if many results match the search but are inaccessible to the user.
- Diff viewer keyboard shortcuts do not work in Opera. In general, Opera is not a supported browser [124]. While most functionality is available, some features, such as keyboard shortcuts do not work.
- When upgrading from 2.1 to 4.0 the server sometimes starts up with a JDBC error. Solution: Check the database and LDAP password fields in `installation-directory/tomcat/conf/Catalina/localhost/ROOT.xml`. Repeated dollar signs ('$'), necessary in PeerReview Complete 2.1, are not required. Remove them and restart.
- Custom reports with a large number active columns or filters; or with long filter text can break the bookmark, SQL, Printable, and CSV links in Internet Explorer. Because the links contain all the filter information, the URL's can exceed the IE's maximum URL length (2083 characters).
  Workaround: Use Firefox for complex reports
- Character set differences that change the location or number of line breaks can change the way comments get promoted.
  Workaround: If necessary, use the auto-detected character set to make comments and mark conversations read.
- Character set differences that change the location or number of line breaks result in metrics that may not be perfectly correct. The metrics for files are computed using the auto-detected character set.
- If there are multiple files with the same filename (but different paths) in a review the automatically linked filenames in review text (custom fields, chat, etc.) may not link to the intended file.

Known Issues with the Command-Line Client [196]

- Case 31758: Can get an `ClassCastException` using the new v2.0 client against a v1.x server with certain commands and with certain v1.x servers.
  Workaround: Make sure your client and server versions match.
- Case 24209, 23958: The system `PATH` variable is not always updated properly by the installer [182], or the `PATH` variable is updated but running applications are not notified, so it takes the user logging out of Windows or even rebooting before the command-line client can be used.
  Workaround [204]: Reboot your machine. Or try just logging out and logging back in to Windows.
- Case 34477: Command-line echo passwords back when you type them.
  Workaround: Use Java v1.6.
- Control Characters in text are replaced with the Unicode Replacement Character. This is most often encountered with smart quotes pasted into changelist descriptions.
- Also see issues for specific version control integrations below.

© 2003-2011 SmartBear Software

Known Issues with the GUI Client 189

- *No known issues at this time*

Known Issues with the Tray Notifier 278

- Dock Icon on MacOS: The tray notifier process creates an icon in the MacOS dock. If you launch the GUI Client it creates another copy of the same dock icon.

Known Issues with CVS Integration 302

- Case 42393: Error initializing Add to Review Wizard occurs with reason `'CVSROOT may only specify a postive, non-zero integer port'`. This is an error reported by the CVS command line client (cvs.exe) when there is an unexpected colon in the the CVSROOT value. The Collaborator client requires that cvs.exe work normally on the repository. Using a different CVS client, such as the Cygwin client, can fix this problem.

Known Issues with Perforce Integration 366

- Case 34890: Files on certain changelists show up as new files -- without a "previous version" diff.
  Workaround: This is known to happen with certain 2005 versions of Perforce. Upgrading to the latest Perforce client fixes this issue.

Known Issues with Subversion Integration 396

- Directory entries are ignored - so adding an entire directory does not show up in a review.
- Symlinks are not supported pre-commit - a Symlink uploaded pre-commit will show invalid content in a review.

Known Issues with Vault Integration 390

- Cases 37383, 42625: Adding changes does not find modified Vault source files, and reports many or no modified files. This is a known issue with the SourceGear Vault client that can be resolved by setting the "Detect modified files using CRC's instead of modification times" in the Vault client options for "Local Files".

Known Issues with Team Foundation Server Integration 349

- Case 42153: "Unable to determine the source control server" error can occur when adding modified files. There must be a corresponding Team Foundation Server working folder for the directory configured in the GUI client, or the directory specified for the command line client to avoid this error.
- Case 42787: Team Foundation Server integration will not work with non-English installations of Visual Studio .NET. Regional settings for other locales are supported, but installing a non-English Visual Studio prevents correct parsing of `tf.exe` output.

8.2 Appendix B: Version History

Get notified automatically when a new version is available!
SmartBear announces new publicly-available versions of PeerReview Complete using a mailing list.

If you want to **sign up to receive e-mails**, you can do so at the SmartBear Software Subscription Center:

© 2003-2011 SmartBear Software

http://www2.smartbear.com/SubscriptionCenter.html

Links to individual versions:

Version 6 463
Version 5 472
Version 4 485
Version 2 502
Version 2 512
Version 0/Alpha 519

8.2.1 Version 6

6.5.6500 — 2011/07/22
*** FIRST RELEASE OF v6.5! ***

Major features:
- Report Access Security 76 features
- Improved file upload support

- added --- New vmoptions to adjust when review auto-update is disabled.
- added --- Users can now add changelists to a review if the regex matches the review (Case 57989)
- added --- Support for Eclipse alternate file systems
- added --- New Report Access Security
- added --- Support for batching XMLRPC requests
- fixed --- Eclipse plugin now correctly referenced in PRC server (case 58623)
- fixed --- Fixed a problem where going to a nonexistent review would create a NPE. (Case 57456)
- fixed --- Converted a help link which opened an XML file to raw text to compensate for browser incompatibility (Case 58547)
- fixed --- Improved error messages if group synchronization fails to parse correctly
- fixed --- Corrected a link on the Admin panel that was opening the wrong help section (Case 57780)
- fixed --- Corrected some online manual links that resulted in 404 errors (Case 57677)
- fixed --- Running 'addgitdiffs' on a bare repo will no longer create a NPE (Case 58240)
- fixed --- Corrected issue with Subclipse and SVN 'mergeinfo' support (Case 57555)
- fixed --- Updated change type in already existing versions for TFS (Case 57371)
- fixed --- Changed diff uploads to use streams
- fixed --- Updated fun facts to be more generic (Case 57972)
- fixed --- Several corrections to the owner's manual
- fixed --- Corrected missing space in 'ccollab addversions' documentation (Case 57964)
- fixed --- Removed request for 'mergeinfo unsupported' error with Subversive (Case 57555)
- fixed --- Corrected '%1 is not a valid Win32 application' error when installing Print To Review (Case 58006)

6.1.6104 — 2011/05/25
- added --- support Subversive 0.7.9.20110207-1700
- fixed --- More logging during server startup
- fixed --- Server triggers replacements failed for custom fields with whitespace (Case 57516)
- fixed --- All Defects report filtered by group did not show defects (Case 57033)

© 2003-2011 SmartBear Software

- fixed --- License expiration date is now always displayed in UTC with clear timezone indication (Case 57473)
- fixed --- TFS changeset deletes not showing as deleted (Case 51361, 57371)
- fixed --- Error processing filenames containing ampersands on windows (Case 57437)
- fixed --- Remove spurious extra logging in Eclipse plugin
- fixed --- Poor write performance when content-cache is located on a network share
- fixed --- Parsing of StartTeam diff filenames with embedded spaces (Case 57197)
- fixed --- Error reviewing subversion revision 1
- fixed --- Parse Surround history output with multiple continuation lines (Case 57288)

6.1.6103 — 2011/04/15

- added --- Installers for 64-bit JVMs on Windows
- fixed --- Print To Review not deleting image files on cancel (Case 54914)
- fixed --- "Download Diffs" fails for large reviews. (Case 56889)
- fixed --- "Download Diff" link now checks SCM permissions. (Case 57110)
- fixed --- "Open in external diff viewer" link now checks SCM permissions. (Case 57110)
- fixed --- Ticket verification should be case-insensitive (Case 57058, 57075)
- fixed --- Eclipse plugin Defect links truncated
- fixed --- Eclipse plugin should post chat asynchronously
- fixed --- Remove ccollab-cvs trigger from installer (Case 56876)
- fixed --- Support SourceGear Vault Professional Edition (Case 56205, 56737, 56863)
- fixed --- Fix Surround 'Archive error: -9014'
- fixed --- "Suppress Notification" checkbox does not stay checked. (Case 56929)
- fixed --- Chat pane doesn't load for binary files (Case 56749)
- fixed --- Support path-based access control for Apache-based Subversion servers (Case 56657)
- fixed --- Support blank option values on the command-line
- fixed --- attachments via web UI fail (Case 56783)
- fixed --- TFS commit issue (Case 56528)
- fixed --- TFS shelvesets author check should be case-insensitive (case 56436)
- fixed --- NPE when file has no base version (Case 56553)
- fixed --- NPE on alternate git status text (Case 56443)

6.1.6102 — 2011/03/09

- fixed --- Server fix for broken `ccollab admin wget` and Eclipse plugin w/ MySQL or Oracle. (Case 56315)
- fixed --- Add product name to password prompt dialog
- fixed --- Fix AppSetting not found error configuring Studio Addin for debug
- fixed --- Overwrite stale ticket in GUI client and Eclipse (Case 56426)
- fixed --- Error committing Subversion file with spaces (Case 56534)
- fixed --- Older (4.0) clients depleted database connection pool. (Case 56568)
- fixed --- Invalid regular expression in ExtJS (Case 55848)
- fixed --- RTC integration missing icon
- fixed --- NPE refreshing Review Summary Page (Case 56462)
- added --- Support MKS Source Configuration Path project types (Case 56327)

6.1.6101 — 2011/02/22

*** FIRST RELEASE OF v6.1! ***

© 2003-2011 SmartBear Software

Major features:

- Tasktop Certified Mylyn Integration
- Ticket-based logins
- Support Subversive Eclipse Subversion plugin

6.0.6018 — 2011/02/18

- fixed --- Don't limit reviews to N changelists (Case 56011)
- fixed --- Eclipse Review Editor errors "Graphic is disposed" (Case 56393)
- fixed --- Attachments should be archived with other clients.

6.0.6017 — 2011/02/11

- fixed --- satisfy TaskTop requirement that the error icon used in the collaborators section of the review editor should be the one commonly used in eclipse
- fixed --- satisfy TaskTop requirement that required custom fields be decorated with a warning instead of an error icon.
- fixed --- LDAP auth with DOMAIN/username broken (Case 55481)
- fixed --- Browser hangs loading native document conversations (.doc file) (Case 56253)
- fixed --- typo in documentation for 'ccollab logout'
- fixed --- TFS deleted files showing as initial (Case 55981)
- fixed --- Case 56253
- fixed --- Explicitly support "M/d/yy h:mm:ss a" when parsing Surround changelists, instead of expecting this to be the default system date format (Case 56079)
- fixed --- AddFiles should not attempt to add a base version (Case 56157)
- fixed --- Don't require Eclipse CVS plugin (Case 56175)
- fixed --- Supply Subversion username and password when adding arbitrary diffs
- fixed --- Allow users to further configure dynamic reports before running them. (Case 54888)
- fixed --- Dynamic reports will show more useful URLs in the URL bar if possible.
- fixed --- Apply and OK buttons disabled in the GUI client (Case 56120)
- fixed --- NullPointerException adding pending changelist with Eclipse plugin (Cases 55998, 56080)
- fixed --- remove extra prompt for password on ccollab login
- fixed --- Rename products in manual.
- fixed --- Don't require access to Subversion repository root (Case 53783)
- fixed --- Don't require access to Subversion repository root (Case 53783)
- added --- Support Vault Professional and 5.1.1 release (Case 56205)
- added --- diagnostic to add users to ALL_USER group if they aren't already in it (Cases 56090, 56101). this is an empty commit because the original commit
- added --- documentation for $review.defectlog variable (Case 56196)

6.0.6016 — 2011/01/19

- added - Respect Subversive Plugin "Do not select externals" preference
- added - Subversion 'svn-recurse-externals' 396 option to recurse in to svn:externals directories
- added - Subversion svn-auto-add 396 global option to treat unversioned files as "added"
- added - Better command-line messages for ccollab admin group member add 276
- added - review_activity_summary 432 columns: active_reviewer_hours, passive_reviewer_hours (Case 55374)
- added - Print To Review 64-bit driver (Cases 55373, 54985, 52098)
- added - Extra logging around search
- added - Support for Team Foundation Server 2010 (Case 54907 and others)
- added - Radio buttons for Clearcase "Add versions..." GUI file selection usability (Case 50471)
- added - Support Subversive Eclipse Subversion plugin

© 2003-2011 SmartBear Software

- added - Diagnostic to clean up duplicate values in metadatavalue* tables (Case 55633)
- added - Prompt from eclipse client and standalone gui when bad auth credentials prevent succesful server responses.
- fixed - Handle TFS Russian date formats (Case 55797)
- fixed - Bad warning message when uploading unversioned file using 'ccollab addfiles'
- fixed - Suppress messages going to Subclipse console
- fixed - Error committing Subversion files in added directories
- fixed - Prevent duplicate Group members (Case 55623)
- fixed - Delete duplicate Group members in Fix Database Data diagnostic
- fixed - Eclipse resource decoration and defect markers fail to update
- fixed - Eclipse "Add to Review" menu item in Synchronize View missing for CVS and Subversion
- fixed - Exception when running "List Reports" in Oracle (Case 55721)
- fixed - GUI Client Git "Add changes ..." doesn't canonicalize filesystem path (Case 54598)
- fixed - NPE in "Add changes..." when adding from subdirectory of Git repo (Case 55676)
- fixed - Handle missing Perforce 'haveRev' better (Case 55458)
- fixed - Reviewer can incorrectly finish until file activity (Case 55456)
- fixed - Reworked files showing 'Initial' status (Case 55557)
- fixed - NPE in installer during ROOT.xml generation (Case 55578)
- fixed - Better error message when LDAP authenticated user hasn't logged in to the web UI (Case 55538)
- fixed - RTC server plugin fails to create review when files paths cannot be constructed (Case 55547)
- fixed - Error saving Review custom fields in Eclipse plugin (Reported in forum)
- fixed - Performance - Only search "user" comments, not system comments
- fixed - Skip change set links that are missing a workspace hint in RTC Server Plugin
- fixed - Translate ClearCase LATEST to actual version
- fixed - Try harder to not run out of memory when running Perforce commands (Case 46879)
- fixed - Tray Notifier always thinks there are new Action Items
- fixed - Can not add pending Perforce changelist from Eclipse Plugin (Case 55064)
- fixed - Improve initial load speed of diff page (Case 55306)
- fixed - Command-line process execution fails with msysgit (Case 55322)
- fixed - Collapse paths in tree when selecting locally modified files in "Add versions" UI (Case 50471)
- fixed - Stop storing password supplied at command line prompt in config file
- fixed --- ClassCastException in "Fix Duplicate Users Differing By Case" Diagnostic. (Case 55910)

6.0.6013 — 2010/11/17

- added --- Support KB, MB and GB suffixes for smartbear.ccollab.upload.truncate.size (Case 54922)
- added --- Eclipse plugin support for reviewing URLs
- added --- Show "pins" in Eclipse Compare Editor for images and documents
- added --- Support for document review in Eclipse plugin
- added --- The assignment_state database view describes actioncodes in the assignment table. (Case 55049)
- added --- Find changes on ClearCase branch for Add Versions (Case 50471)
- added --- Edit participant custom fields in Eclipse plugin Review Editor
- added --- New "ccollab admin review edit --participant-custom-field" option
- added --- Clearcase: UI for add versions from branch (Case 50471)
- added --- Ability to "Poke" a Review participant in the Eclipse Plugin
- added --- Eclipse Plugin Review Editor show participant state using icons and font
- added --- Support Vault 5.0.4/Fortress 2.0.4 (Case 54773)
- added --- Double-click an entry in the Eclipse plugin Review Editor defect log to open the associated file
- added --- Show extended information in Eclipse plugin Review Editor defect log tooltip
- added --- "File" and "Location" columns in Eclipse Review Editor defect log section

© 2003-2011 SmartBear Software

- added --- Specify custom label for conversations in Eclipse Plugin
- fixed --- Throttle error messages about old/unsupported clients. (Case 55194)
- fixed --- Next/prev buttons in diff viewer failed after changing version comparison. (Case 55166, 54870)
- fixed --- NPE when converting null dates to strings (Case 55179)
- fixed --- Initial setup page changed user logins instead of adding new users. (Case 54771)
- fixed --- Support non-English locales with Subclipse integration (Case 55185)
- fixed --- Allow single-word user names (Case 54528)
- fixed --- Version change without content change shows 'Uploaded File' instead of change metrics (Case 55130)
- fixed --- Wrong version uploaded for ClearCase activities with unordered changes (Case 54648)
- fixed --- Change server.xml defaults to improve server behavior under load
- fixed --- Upgrade Tomcat to 6.0.29
- fixed --- NPE updating defect markers in Eclipse plugin (Case 55064)
- fixed --- User selections should include user logins for disambiguation
- fixed --- Links in dynamic reports were broken on first page load. (Case 54934)
- fixed --- Respect custom field phase visibility in Eclipse Plugin
- fixed --- Review overview fails to load - NPE with empty label location on Oracle backend (Case 54839)
- fixed --- Client installer removes other P4V custom tools (Case 54448)
- fixed --- Improve 'Invalid ClearCase version' message (Case 54975)
- fixed --- Visual Studio Addin unloads when solution is closed (Case 52634)
- fixed --- Visual Studio Addin should display error message if nothing to do (Case 54661)
- fixed --- Eclipse plugin mangles multi-line text custom fields (Case 54867)
- fixed --- Wrong base version for Vault rolled back versions (Case 53902)
- fixed --- Eclipse Plugin Review Editor NPE with CodeReviewer (Case 54790)
- fixed --- Prevent MKS host name truncation (Case 53470, 54780)
- fixed --- Prepopulate Print To Review document name (Case 54123)
- fixed --- Defect Custom field order not respected (Case 54717)
- fixed --- P4V integration doesn't prepopulate review title (Case 54449)
- fixed --- New comment text box in Eclipse plugin doesn't expand correctly as you type
- fixed --- Error uploading reverted CVS file in Eclipse plugin (Case 54587)
- fixed --- Be more lenient for administrator full names in support of non-Latin-1 character sets (Case 51433)
- fixed --- Error using relative paths with 'ccollabgui addchanges' (Case 54459)

6.0.6012 — 2010/10/01

- added --- Documentation for content-cache format prop file
- added --- Compare against any local file in Eclipse plugin
- added --- Select line in Eclipse plugin Compare Editor when conversation focused in Conversations View
- added --- Compare against any local file in Eclipse plugin
- added --- "Refresh" button to Eclipse Review Editor
- added --- Display links for URLs and emails in Eclipse Plugin Conversations View
- added --- Links in Eclipse Plugin Conversations View
- added --- Links in Eclipse Plugin custom fields and defect text
- added --- Linkify-as-you-type in Eclipse Plugin
- added --- Display Review ID in "Add To Review" wizard confirmation page (Case 54371)
- added --- Log IP address of old clients (Case 54392)
- added --- New reporting views. (Case 54438)
- added --- Option to suppress notifications by type/template
- added --- Interim 'ccollab logout' command that clears password (Case 54065)
- fixed --- HTML markup displayed in Eclipse plugin 'Moving On' section
- fixed --- Multi-selected files not uploading from addin

© 2003-2011 SmartBear Software

- fixed --- Defect links and comment box sometimes disabled improperly in Eclipse plugin
- fixed --- don't mark review editor dirty just because custom field values are not yet assigned
- fixed --- NullPointerException committing from tray notifier
- fixed --- Eclipse Review Editor "Moving On" section missing controls after pressing "Begin Review"
- fixed --- Make Conversation ruler background color match file in Eclipse Compare Editor
- fixed --- Missing checked graphic in diff viewer
- fixed --- Eclipse plugin Compare Editor "Too Many Differences" error"
- fixed --- Erroneous Concurrency Exception on upgrade (Case 54087)
- fixed --- Can no longer create multiple custom fields that differ only by case (Case 54087)
- fixed --- Corrected features-not-supported-in-CodeReviewer list
- fixed --- Email notifier log message says "seconds" instead of "milliseconds"
- fixed --- Display image for file type in Eclipse Compare Editor
- fixed --- Prefs page fails to render tabs in IE7
- fixed --- In Notification Template admin screen, sort the templates by display name.
- fixed --- Make 'browser' global option case-insensitive (Case 50407)
- fixed --- Keep Eclipse Plugin Action Items View and Editors in sync
- fixed --- Use custom label for "Defect" in Eclipse Plugin
- fixed --- Eclipse plugin "Could not get defect attribute from defect marker" error (Case 54349)
- fixed --- Sort next file/previous file the same as the review overview screen, collating case (case 54132)
- fixed --- p4 \r\r\n line endings confuse ensure-content-reviewed trigger (Case 54213)
- fixed --- Better error messages for Eclipse Plugin (Case 54350)
- fixed --- Don't overwrite file content when restoring from backup (Case 54401)
- fixed --- Installer should automatically replace old P4V custom tool definitions (Case 54450)
- fixed --- Updated manual to document all views. Corrected view name to: defects_by_path (Case 54437)
- fixed --- Dynamic Defect Report filter by Review Completion Date (Case 54226)
- fixed --- Error upgrading server if no diffs cached (Case 54381)

6.0.6011 — 2010/09/08

— FIRST GA RELEASE OF v6.0! —

Note: 6.0 Beta clients (6.0.6001 - 6.0.6009) are not compatible with the GA 6.0.6011 server (5.0 clients *are* compatible with 6.0.6011 server). If you were using a 6.0 beta client please upgrade your client to the GA version 6.0.6011.

- added --- Don't allow saving password via `ccollab set` 215 (use `ccollab login` 214 instead)
- fixed --- Group Admins 91 may delete groups they admin (Case 52208)
- fixed --- Last Comment shows if redacted 173 (Case 53732)
- fixed --- Error entering new Comments on "local file" version in Eclipse plugin (Case 53820)
- fixed --- Create distinct changelists for unmanaged files uploaded via `ccollab addfiles` 217 (Case 52713)
- fixed --- Append indicator page to end of truncated Print To Review 282 documents
- fixed --- Don't allow new conversations when comparing against changedlocal file in Eclipse plugin
- fixed - Eclipse plugin Defect Markers and Label Decorations don't clear when Review deleted
- fixed - Print to Review 282 displays "No Title" error (Case 53929)
- fixed --- NPE on Review Detail Report 179 (Case 53682)
- fixed --- User Reports 180 specify "System Admin" instead of ambiguous "Admin" (Case 53084)
- fixed --- Removed unused 'issue' table (Case 52652)
- fixed --- `ccollab addchanges --diffbranch` 216 option should not return CHECKEDOUT versions (Case 53680)

© 2003-2011 SmartBear Software

6.0.6009 — 2010/08/30

- added --- New file content store format for better FS performance.
- added --- Print To Review and Studio Addin launch 'Add To Review' GUI wizard
- added --- Enable log level management through JMX
- added --- External diff launcher preset for BeyondCompare on Linux
- added --- Supported browsers now include Chrome
- added --- Display graphical UI for P4V integration
- added --- P4V integration upload multiple changelists at once
- added --- Show user login to disambiguate user names on Group admin page (Case 53589)
- added --- Click on a line in the Eclipse compare viewer and start typing to add comment
- fixed --- Added troubleshooting info to the manual for p4v/p4win plugin (Case 53579)
- fixed --- Added keyboard shortcuts for gui client to manual (Case 53578)
- fixed --- Eclipse client new chat box doesn't size correctly with trailing line feed on Windows (Case 52804)
- fixed --- Review completion from the command line ignores unset required participant custom fields. (Case 51714)
- fixed --- Role Configurations cannot be created (Case 53418)
- fixed --- Double-click entry in Eclipse diff's structure view and the window shifts to the right (Case 51590)
- fixed --- SCM Configurations show bogus local path
- fixed --- Install P4 Plugins checkbox should be checked by default if Perforce installed.
- fixed --- Support redacted comment update in eclipse plug-in
- fixed --- Documentation on how to change the file cache location (Case 47974)
- fixed --- Restrict Uploads to Review cannot be changed on review edit (Case 52417)
- fixed --- Show redacted comments as redacted on review overview
- fixed --- Add documentation for configurable server log settings (Case 50107)
- fixed --- Comments sort in wrong order in document review. (Case 53540)
- fixed --- "Fix Defects" phase now uses custom label for Defects.
- fixed --- Eclipse client allows you to "wait" when you're already waiting (Case 51559)
- fixed --- Group member edit list of members is too narrow in IE (Case 50184)
- fixed --- ccollab admin review wait's --until argument is now optional; defaults to Any
- fixed --- Include disabled and reporting-only Groups in Review report filter
- fixed --- Organize Groups on Admin->Groups page (Case 53586)
- fixed --- Change the accepted version icon in the pulldown to differ by more than just color
- fixed --- Display "not yet supported" message for "Live URL" files in Eclipse Plugin
- fixed --- "Concurrency Exception" after system upgrade. (Case 53621)
- fixed --- Select current location when clicked in right side of Eclipse compare viewer
- fixed --- Update top-level Git page in manual (case 53395)
- fixed --- Focus correct conversation control when new conversation started in Eclipse client

6.0.6008 — 2010/08/06

- fixed --- Eclipse update site URL in 6.0 zipped update site pointing to 5.0
- fixed --- Addin under VS 2010 (cases 51282, 52100, and 53275)
- fixed --- Eclipse plugin "Recent Participants" have the wrong Role
- fixed --- Make clearcase addversions wizard upload multiple versions at a time
- fixed --- Restrict Access to Review has inconsistent wording (Case 52444)
- fixed --- Removed broken "Revert" buttons from Group Admin page
- fixed --- PHP not syntax highlighted when opening <?php is not closed (Case 50039)
- fixed --- In document review, clicking on a different page's comment does not select that comment (Case 52470)
- fixed --- Display an error on review edit when not all required assignments are present (Case 52008)

© 2003-2011 SmartBear Software

- fixed --- Diff viewer keyboard shortcuts help doesn't work in Chrome (Case 51820)
- fixed --- Eclipse client custom field error decoration clipped in Review Editor
- fixed --- Eclipse Review Editor appears dirty if string custom field is empty
- fixed --- Group selection does not appear in review editor if no group is assigned (case 52872)
- fixed --- ClearQuest activity review titles should default to headline (case 49180)
- fixed --- Use first commit hash as Git repo identifier
- fixed --- Allow non-ascii characters in username/password field (Cases 51434 and 52406)
- fixed --- Sort files alphabetically in GUI client "Add Changes" wizard (Case 53067)
- fixed --- Typos in documentation of "ccollab admin batch" command (Case 52711)
- fixed --- Redacted comments should not be counted or displayed in reports (Case 52256)
- fixed --- Alphabetize users and groups on Group edit page (Case 53067)
- fixed --- Redacting a comment on zoomed image resizes image (Case 52277)
- fixed --- Spurious P4V addons message at end of install process (Case 48043)
- fixed --- Move restrict process options to access restrictions section (Case 49848)
- fixed --- Error uploading Subversion revisions with directory changes in Eclipse Plugin
- fixed --- Performance - Case insensitive string indexes in Oracle (Case 52392)
- fixed --- Content Archive queries do not scale well
- fixed --- Enforce administrator dump restriction at dump time, not just display time
- fixed --- Surround changelist date AM/PM concerns (case 52522)
- fixed --- Default Value for Participant Custom Field broken (Case 52082)
- fixed --- Changed description of "Allowed to Modify Review" to better reflect true behavior (Case 51954)
- fixed --- When Eclipse plugin refresh job fails with network error, stop all refresh jobs
- fixed --- Order files alphebetically in ClearCase version spec page
- fixed --- Prevent multiple instances of server (multiple ROOT.xml files) from running at the same time
- added --- Button to Eclipse Review Editor to open Review in browser
- added --- Implement defect "Track Externally" button in Eclipse plugin
- added --- User Detail Report now contains a link to get a CSV file containing the User Activity data
- added --- Compare reviewed version with local file in Eclipse client
- added --- 'ccollab admin review defect mark-not-external'
- added --- Option to ignore integration history when caculating previous versions in perforce(Case 51845, 45704)
- added --- Always show line numbers by default in Eclipse plugin line-based Compare Editor
- added --- Print To Review support for 300 DPI color (case 52050)
- added --- Check for concurrent DB access
- added --- Change autofill UI in clearcase addversions wizard
- added --- Instructions in manual on how to create database and user in mysql using command line
- added --- Better error handling for server connection errors in Eclipse plugin
- added --- Submit comments / create defects with in Eclipse plugin

6.0.6006 — 2010/07/09

- added --- Command-line commands address existing Conversations by first Comment ID
- added --- Command-line commands 'ccollab admin review conversation mark-read' and 'ccollab admin review conversation accept
- fixed --- make template editable from eclipse review editor
- fixed --- Defect markers on wrong lines in Eclipse Client
- fixed --- parse diffs files have a Unicode byte order mark (Case 52660)
- fixed --- Enforce administrator dump restriction at dump time, not just display time.
- fixed --- New branched TFS files missing from shelveset review (case 48530)
- fixed --- Better detection of content type in Eclipse compare editor
- fixed --- Better file type icons in Eclipse client Review Editor
- fixed --- Fewer database queries to load diffs on cache miss (Case 52392)

© 2003-2011 SmartBear Software

6.0.6005 — 2010/06/29

- fixed --- Typing in Eclipse Plug-in chat and defect text boxes is slow
- fixed --- Eclipse Plug-in Review Editor icon inconsistent
- fixed --- File URLs break Review overview (Case 51666)
- fixed --- GUI Client missing larger logo sizes
- fixed --- Diff Viewer Prev/Next buttons don't work (Case 52157)
- fixed --- If redacted comment 173 is most recent comment you cannot make the same comment (Case 52224)
- fixed --- Hidden markers 164 shown again when zoom 166 changes
- fixed --- Missing page notification fails to clear after zoom 166
- fixed --- Chat column 167 too narrow in IE8 (Case 52280)
- fixed --- Strange text in confirmation message box when deleting Defect (Case 51460)
- fixed --- User detail report 180 missing one Group (Case 52110)
- fixed --- Javascript error loading Groups admin screen
- fixed --- Participant names are listed in phases where no custom fields are visible (Case 52042)
- fixed --- Eclipse Plug-in Review Editor shows extraneous "treenode" icon on files
- fixed --- Eclipse Plug-in initialize custom fields 104 with default values
- fixed --- Wrong default value for selection custom field
- fixed --- Print to Review 282 installer should update system path variable (Case 52390)
- fixed --- Eclipse Plug-in remove "edit" state for Defects
- fixed --- Unable to parse remote origin url from config file (Case 52589)
- added --- Documentation for Git integration 312 in the GUI Client
- added --- Documentation for redact comment 173 feature
- added --- Command Line Client command "ccollab admin review set-participants 244"
- added --- Documentation for Mercurial 343 integration
- added --- Drag and dropparticipants to change roles in the Eclipse Plug-in Review Editor
- added --- Right-click to edit participants in the Eclipse Plug-in Review Editor
- added --- Display participant custom fields 107 in Review detail report 179
- added --- Eclipse Plug-in UI to add participants
- added --- Eclipse Plug-in Review Editor can drag from users list to participant list to add participants
- added --- Drop-down menus to select which version to diff in Eclipse Plug-in Compare Editor
- added --- Print to Review 282 documentation with troubleshooting info on Windows 7 and Vista (Case 52398)
- added --- UI field assist for custom field 104 errors in Eclipse Plug-in
- added --- Database views 432 for participant custom fields 107 (Case 52324)
- added - Eclipse Plug-in Conversations view can be filtered to show only defects or comments, removed "Defects" and "Comments" views
- added --- Eclipse Plug-in allow multiple instances of Conversations view
- added --- File status (added/modified/deleted) icon decorations in Eclipse Plug-in Review Editor
- added --- Show Eclipse icon for file type in Eclipse Plug-in Review Editor

Fixes from 5.0 build 5033 474:

- fixed --- P4 integration 366 ignores unmodified 'move/add' files when configured to 'revertUnchanged' (Case 52319)
- fixed --- Command Line Client "ccollab addchanges" link in Subversion command line documentation (Case 52413)
- added --- Support adding MKS change packages from other authors (Case 52317)
- added --- Performance improvements for custom field queries
- added --- Option to specify how frequently server checks for stalled reviews (performance) (Case 52392)

Fixes from 5.0 build 5032 474:

© 2003-2011 SmartBear Software

- fixed --- TFS 349 Date parsing error in New Zealand (Case 51476)
- fixed --- Improve performance - remove some wasteful queries

Fixes from 5.0 build 5031 474:
- fixed --- Not finding CMVC 318 track and find binaries on Solaris (Case 51930)
- fixed --- Error adding TFS 349 Shelveset to Review (Case 51386)
- fixed --- Performance issues with SQL Server 27 activity queries (Case 52086)
- fixed --- Not all defect activity pulls defect activity trigger 115 (Case 51898)
- fixed --- Dynamic reports now properly display MSSQL 27/HSQL 25 dates (Case 49695)
- fixed --- Don't include Group User members 92 in configuration-only dump
- added --- More debug logging for "add diffs 200" commands

6.0.6001 (beta) — 2010/06/02

*** FIRST RELEASE OF v6.0 BETA! ***

Major new features include:
- Support for reviewing Microsoft Office documents 282
- An Add-In for Visual Studio 286 - add files to a review from the Solution Explorer
- Significant enhancements to the integrations with ClearCase 322, Rational Team Concert, Git 311, and Mercurial 343
- Ability to redact comments 173

8.2.2 Version 5

5.0.5041 — 2011/02/18
- fixed --- Update Vault 390 5.0 references to 5.1 (Case 56205)
- fixed --- TFS 349 deletes showing up as initial (Case 55981)
- fixed --- NullPointerException adding pending changelist with Eclipse plugin (Cases 55998, 56080)
- fixed --- Backport TFS 349 Russian date fix to 5.0 (case 55797)

5.0.5040 — 2011/01/11
- fixed --- Update Vault 390 5.0 references to 5.1 (Case 56205)
- fixed --- TFS 349 deletes showing up as initial (Case 55981)
- fixed --- NullPointerException adding pending changelist with Eclipse plugin (Cases 55998, 56080)
- fixed --- Backport TFS 349 Russian date fix to 5.0 (case 55797)
- fixed --- Prevent duplicate Group members (Case 55623)
- fixed --- Reviewer can incorrectly finish until file activity (Case 55456)
- fixed --- Translate ClearCase LATEST to actual version
- fixed --- Support deleted TFS files
- fixed --- Error uploading reverted CVS file in Eclipse plugin (Case 54587)
- fixed --- Version change without content change shows 'Uploaded File' instead of change metrics (case 55130)
- fixed --- Wrong version uploaded for ClearCase activities with unordered changes (case 54648)
- fixed --- Improve 'Invalid ClearCase version' message (case 54975)
- fixed --- Wrong base version for Vault rolled back versions (case 53902)
- fixed --- Support Vault 5.0.4/Fortress 2.0.4 (Case 54773)

© 2003-2011 SmartBear Software

- fixed --- Prevent MKS host name truncation (cases 53470,54780)
- fixed --- Updated manual to document all views. Corrected view name to: defects_by_path (Case 54437)
- fixed --- "sort next file/previous file the same as the review overview screen, collating case" (Case 54132)
- fixed --- Prefs page fails to render tabs in IE7

5.0.5039 — 2010/09/17

- fixed --- defect triggers don't fill in substitution for ${actor.*} (case 53618)
- fixed --- '--diffbranch' option should not return CHECKEDOUT versions (Case 53680)

5.0.5037 — 2010/08/25

- added --- Configurable AccuRev history algorithm
- added --- VHDL syntax highlight (Case 53648)
- fixed --- Fix type coercion error in oracle compatibility (Case 52573)
- fixed --- Repair corrupted MetadataValueInteger fieldIds (case 52573)
- fixed --- Do not load inactive custom fields (Case 52573)
- fixed --- Chat Pane in FireFox 3.x too wide in Linux (Case 53580)
- fixed --- Update jPDFImages - PDF rendering never completes - 100% CPU load (Case 53486)
- fixed --- "Download Diff" generates invalid diffs. (Case 53479)

5.0.5036 — 2010/08/10

- fixed --- removed broken "Revert" buttons from the Group Admin page
- added --- Allow user to provide Subversion a HTTPS client certificate password at runtime (Case 53263)
- fixed --- Validate that a changelist can be detached (Case 53375)
- added --- Diagnostic to find and fix reviews that have illegally detached changelists (Case 53375)

5.0.5035 — 2010/07/29

- fixed --- Enforce administrator dump restriction at dump time, not just display time
- fixed --- Error uploading Subversion revision with directory changes in Eclipse Plugin
- fixed --- Performance - case insensitive string indexes in Oracle (Case 52392)
- fixed --- Content archive queries did not scale well
- fixed --- Auto-detect charecter encoding of user-provided diffs instead of using system default
- fixed --- Don't flood output.log with regular log messages
- fixed --- Typos in documentation of "ccollab admin batch" command (Case 52711)
- fixed --- Allow non-ascii characters in username/password field (Case 51434, 52406)
- fixed --- ClearQuest activity review titles should default to headline (Case 49180)
- added --- Option to ignore integration history when calculating previous versions in Perforce (Case 51845, 45704)

5.0.5034 — 2010/07/09

- fixed --- Parse diff files that start with Unicode Byte Order Mark(Case 52660)
- fixed --- Reduce number of queries required to load diffs (Case 52392)
- fixed --- New branched TFS files missing from shelveset review (Case 48530)

© 2003-2011 SmartBear Software

- added --- Server plugin for Rational Team Concert, Beta

5.0.5033 — 2010/06/23
Upgrade to ExtJS 3.1. This fixes many UI issues including:

- fixed --- Diff Viewer resize issues in Internet Explorer (Case 52189)
- fixed --- PeerReview Complete defect form entry display issue (Case 52096)
- fixed --- Participants list boxes when creating/editing review are cut off in ie6 (Case 52032)
- fixed --- Missing scrollbars on initial diff view load (Case 38247)
- fixed --- Image diffs in IE 6 don't work (Case 46236)

- added --- Support adding MKS change packages from other authors (case 52317)
- added --- Option to specify how frequently server checks for stalled reviews (performance) (Case 52392)
- fixed --- P4 integration ignores unmodified 'move/add' files when configured to 'revertUnchanged' (Case 52319)
- fixed --- Updated backup/migration docco to cover how to point collab to a new database (Case 52411)
- fixed --- Performance - Improve custom field queries
- fixed --- Must provide value for sandbox (case 49641)
- fixed --- Fix 'Value requires 2 integer parts' exception (case 51225)
- fixed --- Support uploads of change packages from other authors (case 52317)

5.0.5032 — 2010/06/14
- fixed --- Don't include group user members in configuration-only dump
- fixed --- Performance - Remove some wasteful queries.
- fixed --- ClassCastException in Oracle data loading code.
- fixed --- TFS Date parsing error in New Zealand (Case 51476)

5.0.5031 — 2010/06/10
- added --- More debug logging for "add diffs" commands (Case 52124)
- fixed --- Wrap long error messages (Case 51915)
- fixed --- Not finding CMVC 318 track and find binaries on Solaris (Case 51930)
- fixed --- Error adding TFS 349 shelveset to review (Case 51386)
- fixed --- Performance fixes for SQL Server activity queries (Case 52086)
- fixed --- Not all defect activity runs defect activity trigger (Case 51898)
- fixed --- Dynamic reports now properly display MSSQL/HSQL dates (Case 49695)

5.0.5030 — 2010/05/22
- added --- Support for 'addactivity all' (Case 50444)
- added --- Better documentation for addversions command.
- added --- option to specify local database path in Rational Synergy (Case 51587)
- added --- Forward compatibility; improve error messaging when connecting to a 6.0 server.
- fixed --- Addactivity hangs reading activity (Case 51276)
- fixed --- ClearCase performance improvements (Case 50942)
- fixed --- Incorrect timezone offset for submitted Perforce changes (Case 50679)
- fixed --- Create new User Account subtext (Case 51328)
- fixed --- Handle deleted files in ClearCase activities (Case 51275)
- fixed --- Reports CSV output should not consume a license (Case 51457)
- fixed --- Auto-refresh clears unsubmitted comments in the text area (Case 51369)

© 2003-2011 SmartBear Software

- fixed --- Reviews created with addversions include files from other reviews (Case 50682)
- fixed --- Notification emails not being sent. (Case 51204)
- fixed --- Command line client missing edit file list prompt message.
- fixed --- NPE when content unavailable; blank screen loading user home page.

5.0.5029 — 2010/04/20
- fixed --- Add per-file lines metrics to review-xml (Case 51010)
- fixed --- Some PDFs don't show up in reviews (Case 50826)
- fixed --- Performance improvement - fewer queries to load action items.
- fixed --- Performance improvement - fewer queries to perform comment promotion; render review overview
- fixed --- Performance improvement - reduce computation when rendering review overview
- fixed --- Performance improvement - reduce computation required to map from SQL result sets to datamodel objects.

5.0.5028 — 2010/04/13
- added --- Technology Preview of IBM Rational Team Concert support
- fixed --- Limit syntax highlighted file size to prevent CPU spike when processing large files(Case 51015)
- fixed --- 'not a ClearCase object' error on Linux (case 50938)
- fixed --- Synergy passwords in clear text of debugging logs (Case 50925)
- fixed --- Fix NPE in logging (case 50799)

v5.0.5027 — 2010/03/26
- added --- 'ccollab admin review defect delete'
- added --- 'ccollab admin review defect mark-fixed' and 'ccollab admin review defect mark-opened' commands
- added --- "ccollab admin review edit" "--deadline" option to change Review deadline (Case 50739)
- fixed --- Bad data on 'ccollab admin review-xml' in some cases using Sun JRE 6. (Case 50372)
- fixed --- Setting multi-valued fields from 'ccollab admin review defect edit'
- fixed --- Can't Accept or comment after a file has been deleted (Case 50466)
- fixed --- Manual did not reflect changes to review deadline implementation (Case 47876)
- fixed --- Bad error message when command-line scm autodetect fails
- fixed --- complex histories cause poor performance adding files with Synergy (Case 50594)
- fixed --- error creating new custom field with same name as deleted custom field (Case 50457)
- fixed --- Improved performance of 'ccollab addchangelist' with multiple changelists
- fixed --- strip xml1.0 unfriendly chars from arguments list of xmlrpc calls
- fixed --- addactivity diffintegration option diffs wrong integration stream version (case 49870)
- fixed --- Set default review deadline when creating reviews from command-line (Case 50739)
- fixed --- better debug logging for "ccollab admin wget"
- fixed --- 'ccollab admin wget' fails with servers not at root of host (Case 50101)
- fixed --- NPE in logging when log message is null (Case 50799)

v5.0.5026 — 2010/03/03
- added --- Command-line support for creating Defects and Comments at page coordinates
- added --- Documentation for configurable cache settings. (Case 50104)
- added --- Support Perforce Eclipse plugin v2009.2.234487
- added --- Allow setting of a review's 'display changelists as' setting from the commandline (Case 50327)

© 2003-2011 SmartBear Software

- fixed --- Toolbar titles misaligned (Cases 48690, 48846, 49976)
- fixed --- Preserve state of Accept and Mark Read buttons after canceled defect edit. (Case 49512)
- fixed --- HTML encoding interacting with auto linking. (Cases 49794, 50010)
- fixed --- StarTeam add-diffs 225 not capturing path info (Case 49875)
- fixed --- Can't cancel, externalize, or edit Defect on Review overview (Case 47734)
- fixed --- NullPointerException in DocumentPageServlet when engine is not available (Case 50134)
- fixed --- Updated manual with current MySQL installation instructions 25 using the supported tools (Case 50002)
- fixed --- Updated docs with information on integrated authentication with SQL Server 27 (Case 49853)
- fixed --- NPE when version server path empty (Case 49944)
- fixed --- Race condition between content archiver and review creation (Case 30131)
- fixed --- Invalid cached differences result in no diffs being displayed (Case 30131)
- fixed --- Paths in reports are double encoded
- fixed --- Poor error behavior when "p4 set" and "p4 info" have no output (Case 50331)
- fixed --- Wrong ClearCase 322 branch predecessor for zero versions (Case 49102)
- fixed --- MKS 358 change packages should always be pending (Case 48754)
- fixed --- Updated manual to clarify overdue deadline status messages

v5.0.5025 — 2010/02/10

- fixed --- Background services not started if database was unavailable at startup
- fixed --- Bad error message when no assigned tasks are found (Case 49423)
- fixed --- Can not add yourself as a participant to an existing review (Case 49419)
- fixed --- Default review title not set for addactivity 229 (Case 49180)
- added --- Getting started links on the last page of the client installer 183
- added --- 'ccollab admin wget' 237 command
- fixed --- Real paths for Rational Synergy 336
- fixed -- The "Link" link in dynamic reports 177 now works again. (Case 49199)
- fixed --- MKS 358 files incorrectly marked as new (case 48775)
- fixed --- Canonicalize MKS 358 host names (case 49466)
- added --- Keyboard shortcuts in the GUI client 189
- fixed --- Verify connection to server when setting up Subversion 399 (Case 49773)
- fixed --- Updated installation documentation 27 for SQL Server 2008, Java 6, and JDBC drivers (Case 49238)
- fixed --- Users duplicated in drop-downs when using Groups (Case 49594)
- added --- Update browser launcher to support GNOME open, KDE open, and XDG open (Case 49677)
- fixed --- Participant filter name in dynamic reports is once again "participantLogin". (Case 49584)
- added --- 'ccollab commit --dismiss-only' 232 option

Fixes from 4.0 build 864 485:

- fixed --- List report 181 results improperly cached (Case 49572)
- fixed --- LDAP lockout can occur if PeerReview Complete is used infrequently. (Case 49526)
- fixed --- "Too many open files" error in Content-Cache Diagnostic (Case 49687)

v5.0.5024 — 2010/01/14

- added --- External Diff Launcher variable "after.version.localFilePath" to enable launching editor (http://uservoice.com/a/7wS9r)
- added --- JMX Interface to Diagnostics
- added --- addversions support for Rational Synergy (Case 48922)
- added --- edit defects on the command line (Case 49142)
- fixed --- filtering a report based on users now works properly with multiple participants in each role
- fixed --- NPE getting content from deleted file using ccollab adddiffs with before and after directories
- fixed --- Visual Studio Addin doesn't size correctly (Case 48914)
- fixed --- Visual Studio Addin should show wait cursor while busy (case 48914)
- fixed --- option to start Synergy session with -rc (Case 48922)

© 2003-2011 SmartBear Software

- fixed --- Perforce local changelists not showing reverted files as reverted (Case 48321)
- fixed --- unrepresentable characters break RPC interface (Case 46761)
- fixed --- Sort action items consistently
- fixed --- VSS diff anomalies (case 47484)
- fixed --- Review too large for refresh algorithm is wrong
- fixed --- use Eclipse proxy settings UI, if available (Case 41725)
- fixed --- broken error message on custom fun facts.
- fixed --- Copy-paste not working in web diff viewer, selection is always cleared(Case 49193)
- fixed --- NPE when completing add changes wizard on review selection screen
- fixed --- email notifications stop being delivered (Case 45913)
- fixed --- NPE in Starteam diff support (Case 49263)
- fixed --- Clearcase '--diffbranch pre' option picks wrong predecessor (Cases 49102, 48802)

v5.0.5023 — 2009/12/11

- added --- Better debug logging for add*diffs commands (Case 48205)
- added --- "ccollab addfiles" option --relative-to (Case 46457)
- added --- StarTeam '--stcmd-exe' option (Case 48622)
- added --- Diagnostics section to the manual (Case 48571)
- added --- High Availability Best Practices section to manual
- added --- Content store cleanup diagnostic
- fixed --- Display external diff launcher errors on MacOs
- fixed --- Custom fields values reset when non-participant adds self to a review (Case 48338)
- fixed --- Support p4 2009.1 move/add and move/delete actions (Case 48312)
- fixed --- Default external diff viewer doesn't launch on Mac OS (Case 48356)
- fixed --- Review dump not available from searching by review ID
- fixed --- Do not install AJP connector by default. (Case 48372)
- fixed --- Automatic session creation for Rational Synergy (Case 48065)
- fixed --- 'specify p4 command Charset' needs to be a boolean drop down in the web UI (Case 48071)
- fixed --- .smartbear directory too accessible (Case 48001)
- fixed --- Handle Synergy objects not associated with tasks (Case 48623)
- fixed --- Subsequent MKS sessions fail to connect (case 48642)
- fixed --- Action Items refresh should cache login for 20 minutes
- fixed --- Require non-blank label for "Group" (Case 47463)
- fixed --- Require non-blank external URL (Case 47351)
- fixed --- Require non-blank Group title (Case 47487)
- fixed --- Require non-blank Group title (Case 47487)
- fixed --- Ignore unmerged branches in history (Case 48623)
- fixed --- Error in creating Automatic Link removes creation data (Case 47616)
- fixed --- Notifications need not refer to existing reviews for Database Integrity Check to pass (Case 48688, 48727)
- fixed --- Determine perforce client is unknown when 'p4 info' output gives no client root (case 48525)
- fixed --- NullPointerException in shutdown if datamodel could not be initialized.
- fixed --- Security fix for XSS vulnerability (Case 48559)
- fixed --- Install4j not installing non-gui symlinks(Case 48731)
- fixed --- Added a workaround for SWT/GTK bug (Case 48377)
- fixed --- Unparseable surround date format (case 48762)
- fixed --- Empty files being written to content cache (Case 48834)
- fixed --- Respond better to coming online before the database after reboots

v5.0.5022 — 2009/11/12

- added — Accessibility improvements for diff view and review overview.
- added — Technology Preview of Rational Synergy support
- fixed — Error message on client when error uploading content to server (Case 48135)

© 2003-2011 SmartBear Software

- fixed — Log error message in server when content-cache is full (Case 48135)
- fixed — Older Perforce servers don't include "SubmitOptions" in client spec (Case 48100)
- fixed — Fix for MKS keyword expansion (Case 47784, 47251)
- fixed — Workaround for 'tf history' bug in Visual Studio 9.0 (Case 47438)
- fixed — Spaces in custom field names broke variable substitution (Case #48105)
- fixed — Log exceptions closing files in content store
- fixed — Make preferences directory as necessary to save "last" review and defect (Case 48194)
- fixed — Old p4 clients ignore -ztag with the where command (Case 48154)
- fixed — Auto-refresh disabled popup is to obtrusive. (Case 48177)
- fixed — Continue if missing MKS revision encountered in change package
- fixed — Rename "Subversion Server GUID" to "Subversion Repository UUID" in UI and Docs

v5.0.5021 — 2009/10/28

- added — log author prep time while in planning phase (Case 47593)
- fixed — Check for compatible product version before trying to restore from a dump file (Case 47869)
- fixed — Upgrade to JavaMail for defect causing email notifications to stop (Case 45913)
- fixed — Report filters display incomprehensible error messages (Case 47768)
- fixed — server should install license file with the EULA (Case 47946)
- fixed — Handle Perforce SubmitOptions: revertunchanged (Case 47488)
- fixed — Remove "Restore Defaults" button from Server Connection preferences page (Case 47950)
- fixed — Error adding a Subversion revision with Eclipse plugin
- fixed — Typo in the mandatory subscriptions error message (Case 47469)
- fixed — Don't specify "-Q utf8" when the customer specifies "p4commandcharset none" (Case 47032)
- fixed — Unrecognized TFS date format for zh_TW locale (case 47813)
- fixed — Spurious P4V addons message at end of the install process on when Perforce is not installed
- fixed — Updated jdpdfimages binaries to fix null pointer when uploading PDF and subsequent PDF rendering problem (case 47879)

v5.0.5020 — 2009/10/15

- added — log author prep time while in planning phase (Case 47593)
- added — Added information about how groups are stored in the database for custom reporting
- added — Support for MKS renames (Case 47037)
- fixed — next/prev file in diff viewer algorithm wrong (Case 47406)
- fixed — review custom field visibility by phase not editable (Case 47283)
- fixed — Diff download format broken when change occurs at end of file.
- fixed — Diff download fails when insertions at end of file. (Case 47748)
- fixed — improve explanation of managing Groups manually vs with sync
- fixed — LDAP access broken when server returns absolute names (Case 47391)
- fixed — Verify Database Schema diagnostic shows bogus errors on Oracle (Case 47732)
- fixed — LicenseDecodingException experienced when system attempts to retrieve stored license (Case 47690)
- fixed — Don't let user install server on Linux if port is in use
- fixed — Auto-refresh of review overview screen locks the browser UI thread (Case 47740)
- fixed — External Diff Launcher fails for certain files on Windows (Case 47649)
- fixed — Edit defect causes tab to be inaccessible to screen reader
- fixed — Error deleting symlink in perforce (Case 47327)
- fixed — Make Add SCM Configuration resize more nicely (Case 47290)

v5.0.5019 — 2009/10/01

- added — DateTime? control in Subversion update changes by date GUI dialog
- added — Database schema diagnostic
- added — Support commit info on changelist current versions
- fixed — Layout of error popup on Eclipse 3.5 and GUI client

© 2003-2011 SmartBear Software

- fixed — GUI Client and Tray Notifier won't launch on 64-bit MacOS (Cases 47336, 46337, 47149)
- fixed — Poor behavior in notifications with anonymous defects (Case 47405)
- fixed — Duplicated output on adddiffs (Case 47409)
- fixed — Missing arg in versionCreate: support commit info for current versions in changelists (cases 44540, 45620)
- fixed — Performance fixes. Reduce total database load when displaying reviews
- fixed — sitting on the review overview page can result in consuming a license (case 46356)

v5.0.5018 — 2009/09/17

- added — Tray Notifier on MacOS (Case 38167)
- added — GUI client and Tray Notifier support for Solaris on X86 (Case 46627)
- added — GUI client support for MacOS 64 bit (Case 46337, Case 47149)
- added — Add a general comment when a changelist is removed from a review (Case 46562)
- added — Make cache performance information available in System screen (Case 47315)
- fixed — 'ccollab admin group create' errors with "Group GUID '...' must exist before you can edit it." (Case 47166)
- fixed — 'ccollab admin group' commands error with "Group GUID '...' must exist before you can edit it." (Case 47166)
- fixed — Added documentation for Review Deadline to the Creating a review section and edited the Creating a review section (Case 47184)
- fixed — The link inside of the GIT integration had a typo that was making it appear malformed (Case 47191)
- fixed — Need to fix the AccuRev casing in the standalone gui (Case 47247)
- fixed — Surround missing from scm config list for CLI
- fixed — Typo on the triggers page in the command for passing a review title (Case 47241)
- fixed — Better explanation of "Active Users" chart (Case 47316)
- fixed — Client doesn't open browser on MacOS (Case 47051)
- fixed — Improve performance of Fun Fact total review time query.
- fixed — Minimum required Surround client is 2009.1.0
- fixed — Better description of review deadline meaning (case 47239)
- fixed — GUI client doesn't close properly on MacOS X (Case 46061)
- fixed — Tray Notifier not resizable on Linux (Case 36320)
- fixed — Diff cache misses almost always (Case 47315)

v5.0.5017 — 2009/09/01

- added — Support Perforce eclipse plugin 2009.1.209672 (Case 47023)
- added — Accessibility improvements to chat, Comment and Defect tabs 167
- added — Allow expansion of changelist title 153 to the full list of included changelists (Case 46928)
- fixed — Subscription 133 authors not displayed when subscriptions are not editable (Case 46423)
- fixed — Diff viewer 157 screen blank for some files - ArrayOutOfBoundsException on server (Cases 46500, 46737)
- fixed — Review report idle times negative on HSQL (Case 46159)
- fixed — Binary files opened from chat links 117 do the wrong thing (Case 46808)
- fixed — Remove incorrect references to "side-by-side" from diff viewer preferences 159
- fixed — Truncate length of large changelist comments before uploading
- fixed — Users are not sorted in participants selection dropdowns 142 (Case 46594)
- fixed — Review Detail Report 179 includes redundant comments (Case 46183)
- fixed — "null" showing up in version selection (Case 46876)
- fixed — SVN executable "svn" is reset when the GUI client 189 restarts
- fixed — Subversion Eclipse plugin support recognize status "Incomplete" (Case 46930)
- fixed — Prevent Tomcat from filling up the output.log file
- fixed — More consistent layout for binary files (non-PDF)
- fixed — Modified Vault 390 files not found (Case 46754)

© 2003-2011 SmartBear Software

- fixed — Modified MKS 358 files in pending change packages compare to wrong predecessor (Case 46511)
- fixed — Miscellaneous Surround 419 fixes (Cases 46348, 45669)
- fixed — Some modified Vault 390 files not being found (Case 46754)

v5.0.5016 — 2009/08/13
- fixed — Handle errors in Subversion 396 working copy (Cases 46307, 35884)

v5.0.5015 — 2009/08/11
- added — Links for further information in the installer 31
- added — Vault 5.0 and Fortress 2.0 390 support (Case 46586)
- added — Support configurable image file types 83 (Case 46632)
- added — Perforce changelist renumbering 366 script to manual (Case 41154)
- fixed — Use configured label for Group 75 in participants error message
- fixed — Scrub invalid dropdown values during upgrade
- fixed — Subversion integration 396 should be case-insensitive on Windows (Case 46561)
- fixed — Installer 31 crash due to malformed xml (Case 35977)
- fixed — Don't count sysadmin on license screen with LDAP 50 (Case 45222)
- fixed — Ignore CVS 302 'no longer in the repository' errors (case 46519)
- fixed — Handle empty Subversion 396 comments (Case 46603)
- fixed — Deleted PDF 164 in changelist causing upload failure (Case 46574)
- fixed — Upgrade fails if no diffs cached in database
- fixed — Better XPath 444 output, including line feed between multiple results (Case 43498)
- fixed — Incorrect timezone application in customizable reports 177 on MySQL (Case 45687)
- fixed — Reviews by Changelist and Unreviewed Changes reports 181 broken if filtered on Changelist ID (Case 46534)
- fixed — Use configured SCM options in "Add Diffs..." button on GUI Client (Case 45436)
- fixed — Log more info when sending email fails (Case 46728)
- fixed — Surround 419 fixes for added files, parsing 'sscm ls' output, and embedded lines in comments (Cases 46480, 45669)
- fixed — Inactive users whose logins are duplicates of active users could block database upgrade (Case 46745)

v5.0.5014 — 2009/07/31
- added — Documentation of "--scm none" option (Case 46456)
- added — Button to restore default notification templates and add missing ones
- added — Button to restore defaults on External Diff preference page (Case 46512)
- fixed — Comments on line 0 cause exceptions (Case 46332, 46379)
- fixed — Expose last activity time in review-xml. This is the review completion time for complete reviews (Case 46467)
- fixed — Participants warning appears when group security is not enabled (Case 46157)
- fixed — Old clients added System Adminstrator to review when adding a changelist whose author did not match a Collaborator user (Case 46187) (server-side fix)
- fixed — Surround SCM committed changelists support and misc fixes (Cases 45669, 45823, 45771)
- fixed — Better logging for login errors (Case 46463)
- fixed — More lenient check for loopback address in p4port
- fixed — Don't autodetect SCM system when printing command-line help usage (Case 46353)
- fixed — Don't create new review for add*diffs commands if no modified files found in diff
- fixed — Cleaner error message for add*diffs command when no diffs found (Case 46135)
- fixed — Display errors returned from 'ccollab addp4diffs' command (Case 46096)
- fixed — Handle odd userId cases correctly (Case 46523)

v5.0.5013 — 2009/07/28

© 2003-2011 SmartBear Software

- added — "Test Connection" button for server-side version control server entry
- added — Create server-side version control server entries automatically from client uploads
- fixed — Error logging status of Subversion connection on server (Case 46249)
- fixed — update-changelist scm trigger finds changelists from a different server

v5.0.5012 — 2009/07/24

- added — Ability to disable "attach changelist" in browser per server-side version control configuration
- added — Support Objective-C syntax highlighting in .m/.mm files (case 46057)
- fixed — Multiple concurrent database upgrade attempts when the Upgrade button is pressed repeatedly
- fixed — potential temp value collision in upgrade could cause all of one dropdown value to be assigned to an other.
- fixed — Error with "ccollab adddiffs new <before dir> <after dir>" with deleted directory (Case 46239)
- fixed — Don't query for database version all the time. (Case 46241)
- fixed — "adddiffs" doesn't roll up with "addchanges" when no SCM system (Case 45964)
- fixed — If using 'p4-protects-script' don't require connection to Perforce Server
- fixed — Support MKS variants in change package files (case 46212)
- fixed — handle anonymous commits in Subversion (Case 46393)
- fixed — User prefs review subscription form field title "Review Creator" is misleading (Case 46310)

v5.0.5011 — 2009/07/20

- added — supply P4PORT to 'p4 -s protects' script
- fixed — Upgrades of SSL Connectors
- fixed — require P4PORT to be an external server name, not "localhost"
- fixed — Improve behavior of activity update queries.
- fixed — Change diff caching mechanism to take load off database.
- fixed — put on schema blinders for oracle indices, triggers, and sequences

v5.0.5010 — 2009/07/16

- added — Option to update ClearCase snapshot views (Case 45296)
- added — Button to accept self-signed Subversion server certificates (Case 46133)
- added — Server-side version control templates
- fixed — working of diff download tooltip on Review overview screen (Case 45985)
- fixed — tray notifier tooltip doesn't refresh (Case 46038)
- fixed — improve line wrapping algorithm (Case 45959)
- fixed — search for filenames fails to return results. (Case 46067)
- fixed — Command line client throws exception if user cancels (Ctrl-C) when prompted for password.
- fixed — scm triggers use server-side version control mapping to find correct server (Case 45841)
- fixed — scm triggers print extra output (Case 45963)
- fixed — groups created in PeerReview Complete enforced after switching to CodeReviewer license (Case 46040)
- fixed — recent Participants link does not work for users not in the group (Case 46055)
- fixed — Improve file headings; especially for diffs.
- fixed — Updatechangelist trigger shouldn't touch changelists descriptions that don't involve the trigger
- fixed — Update perforcetrigger usage message to indicate that it's deprecated.
- fixed — Don't munge whitespace in p4 specfiles when we modify them (Case 46088)
- fixed — Error in Eclipse plugin "fix configuration" action item (Case 46164)
- fixed — SQL Server date format not specified. (Case 44329)
- fixed — new trigger algorithm backwards-compatible with old clients when there's only one Perforce server
- fixed — Improve performance of some queries under Oracle

v5.0.5009 — 2009/07/07

© 2003-2011 SmartBear Software

- updated — jPDFImages library to version 2.13.
- updated — Tomcat to version 6.0.20
- added — Perforce 'Server Address' for version control identification/matching.
- fixed — support for http.nonProxyHost (proxy exceptions) (Case 45855)
- fixed — Check that database supports views before attempting to create views. (Case 45616)
- fixed — Interpretation of some svn output is broken (Case 45775)
- fixed — Wrong Vault predecessor for modified files in VSS mode (Case 45839)
- fixed — Enforce CodeReviewer database restrictions (Case 41184)
- fixed — Enforce CodeReviewer LDAP restrictions
- fixed — Cannot open .zip file from file download (Case 45903)
- fixed — Remove administration screen for configuring Legacy GUI Client versions; it's not supported in 5.0.
- fixed — Support Perforce multi-file diff format in ccollab adddiffs (Case 45420)
- fixed — Suppress 'p4 info': Client unknown warning (Case 45963)
- fixed — Idle users being counted as logged in (Case 46015)
- fixed — Support Perforce client specs with exclusionary mapping rules (Case 45986)

v5.0.5008 — 2009/06/29

- added — Reminder for administrators to migrate from the trial database
- added — Link defects in the defect reports
- added — Syntax highlighting for Ruby
- added — Configure P4V/P4Win integrations in the installer
- fixed — Support for SVN checkouts at drive roots
- fixed — Can't find Accurev workspace (Case 45645)
- fixed — Periodically checkpoint HSQLDB
- fixed — "Update From Smart Bear" sends the user-provided company key
- fixed — NPE in comment promotion
- fixed — External diff doesn't work with too many files (Case 45667)
- fixed — Improve UI for removing a disabled user from a review (Case 45755)
- fixed — Error uploading TFS shelvesets with new files (case 45709)
- fixed — Handle MKS change packages in submitted state (case 45631)
- fixed — No scroll bars in chat window for binary files (Case 45519)
- fixed — AccuRev NPE in ccollab addstream (Case 45764)
- fixed — Better error message for Perforce authentication problem (Case 45778)
- fixed — Incorrect activity start dates (Case 45759)
- fixed — Selecting a marker should scroll it into view (Case 45265)
- fixed — Upgrade 500 to 501 bug on MySQL
- fixed — Upgrade speed fix
- fixed — License check causing spurious log messages.
- fixed — Once disabled, role cannot be reenabled (NPE) (Case 45845)
- fixed — NPE in "Moving On" section (Case 45696)

v5.0.5007 — 2009/06/15

—- FIRST GA RELEASE OF v5.0! —-

- added — Reporting of slow queries in the server log to help diagnose slow operations
- fixed — Slow browser response on some reviews (Case 44872)
- fixed — NPE and divide by zero in client commit (Cases 45480, 45497)
- fixed — NPE in comment promotion
- fixed — Updated HSQLDB to 1.8.0.10 to fix possible data corruption issues with embedded database.
- fixed — Changelists from different SCMs must be kept separate everywhere. (Case 45234)
- fixed — Files not showing up in reviews; TFS versions from early beta clients causing issues (Case

© 2003-2011 SmartBear Software

45504)
- fixed — Changelist rollup should be based on SCM config, where available (Case 45338)
- fixed — group not assigned when review created by client (Case 45511)
- fixed — j_security_check login bug (Case 42864, Case 45102, probably more)
- fixed — Long version names (from ClearCase) cause display issues in diff viewer (Case 45261)
- fixed — Misleading error message on ccollab admin trigger create-review (Case 45540)
- fixed — Verify for P4 2009.1: "p4 info" now reports the server address (Case 45635)
- fixed — create-review trigger doesn't set "last" review variable (Case 45437)
- fixed — Database dump fails when review contains URL versions (Case 45575)
- fixed — NPE when trying to archive files.

v5.0.5005 (beta) — 2009/06/04
- (fixes from release 4.0.860 485)
- (fixes from release 4.0.859 485)
- updated — Manual updated with various new features.
- added — Filters for users on the user administration screens
- added — New variable substitutions: defect.isexternal, defect.externalname, review.group.title, and review.group.guid.
- added — Side by side document review/compare menu working (Case 44443)
- added — MKS change package and addversions support (Cases 43812, 44416, 44540)
- added — User preference for tab width (Case 40604)
- fixed — Company contact information not correct in license files.
- fixed — Improved page load times for review overview page.
- fixed — Improved performance of database query builder (Case 45029)
- fixed — NPE when changing general settings in reviewer.
- fixed — Review list report displays bogus warning when number of rows exactly equals capped number (Case 44525)
- fixed — URL's with embedded credentials not properly linked (Case 44958)
- fixed — create-review trigger argument --review-id-regex <value> should be optional (Case 45091)
- fixed — Compare list/diff viewer headings need more accurate information when using diffs (Case 44672)
- fixed — Next file order does not match compressed tree or tree views (Case 45098)
- fixed — Misleading label for new chat area when reviewing documents or images.
- fixed — Can't jump back to current conversation when on a different page of a document (Case 44229)
- fixed — NPE in GUI client (Case 44914)
- fixed — Line numbers lost for comments on unchanged deleted files (Case 44819)
- fixed — NPE when reverting an unsupported file (Case 45092)
- fixed — Separate multiple changelist description text for readability (Case 44677)
- fixed — Error messages not cleared in attach changelist/url/file dialogs (Case 44726)
- fixed — Firefox sometimes incorrectly guesses RSS encoding (Case 45129)
- fixed — addgitdiffs command now gets more information from git diffs
- fixed — Unable to remove last item from a multiselect custom field (Case 44379)
- fixed — Added command to support Perforce changelist renumbering (Case 44978)
- fixed — Restrict access to fix defect global option does not allow defect creator to fix (Case 45202)
- fixed — Chat box opens for wrong line of code (Case 45149)
- fixed — Support ClearCase 7.1 and int'l date formats (Cases 44914, 45122)
- fixed — Added user state tooltip in the participants section of review overview (Case 43258)
- fixed — addchangelist with Perforce not uploading correct base content (Case 44577)
- fixed — Add configuration for server logging.
- fixed — Workaround for pool users not available.
- fixed — Home page tab counters not updated when content changes (Case 45238)
- fixed — Changed default poke notification text (Case 45258)
- fixed — Corrected timezone for date displays in tables such as the User/Admin screen

© 2003-2011 SmartBear Software

- fixed — String replacements fail with $ (Case 45353)
- fixed — Support multi-line environment variables (Case 43002)
- fixed — User subscription should not (appear to) be pre-populated with the first entry (Case 45317)
- fixed — NPE in content cache diagnostic (Case 45376)
- fixed — multi-select two panel filter doesn't display correctly on IE (Case 44352)
- fixed — respect notification limit in RSS on embedded database (Case 45129)
- fixed — Text from previous review can appear on new review screen (Case 45324)
- fixed — AccuRev getting incorrect previous version (Case 45257)
- removed — Old-style defect reports -- use new defect reports instead. (Case 45215)

v5.0.5004 (beta) — 2009/05/14

- added — Support addversions for Perforce SCM
- (fixes from release 4.0.858 485)

- fixed — Cannot save General Settings in 5.0 (Case 44759)

- fixed — Update documentation for HTTPS configuration
- fixed — duplicate "Adding file" messages with "ccollab addchanges"
- fixed — Remove extraneous "Checking if file exists" messages printed to console with "ccollab"
- fixed — Defects can be added more than once (Case 42403)
- fixed — TFS autodetect causes error when TFS is installed and a file is unmanaged
- fixed — Upgrades fail against SQL Server 2000 (Case 44527, 44730)
- fixed — Next unread comment and next defect buttons do not always work (Cases 41695, 44339)
- fixed — FileMetrics for documents should include page count, but not line counts.
- fixed — Document extension matching (.pdf) is case sensitive (Case 44732)
- fixed — Document review should be unavailable in CodeReviewer
- fixed — Turkish locale issue with capital i (Case 44763)
- fixed — Large changed regions in over under have strange scrolling behavior on line click (Case 46432)
- fixed — Can create defect without supplying required custom fields (Case 40500)
- fixed — Status in review materials is wrong for added files (Case 44839)
- fixed — Command line SCM tokens were case sensitive (Case 44618)
- fixed — Bad VSS login argument on ss.exe command line (Case 44745)
- fixed — Line number validation fails in binary file and URL review (Case 44441)
- fixed — Support for non-ascii filenames in Subversion (Case 43430)
- fixed — Java syntax highlighting rules don't include byte and short
- fixed — Change MIME type of external diffs to be vendor specific (application/vnd.smartbear.cc-diff)
- fixed — URL and document review pane include unnecessary line difference pane (Case 44348)
- fixed — Post-commit review show rework when there was no rework
- fixed — Update documentation on trigger command line syntax (Case 43226)
- fixed — Email addresses not being properly converted to links
- fixed — Bogus error messages in admin screens when inputs are unsaved (Case 40999)
- fixed — Surround password showing up in cleartext (Case 43226, 44954)
- fixed — Custom report filters for meta-data drop-downs need to show disabled items (Case 44504)
- fixed — Diff viewer keyboard shortcuts dialog fails in IE (Case 44338)
- fixed — ArrayIndexOutOfBoundsException when creating svn client config (Case 44961)

v5.0.5003 (beta) — 2009/04/20

- added — refactor review-xml to handle line number/location (Case 44613)
- (fixes from release 4.0.857 485)
- fixed — Make the timezone drop-down items findable and readable.
- fixed — reverting on group edit page displays "updated" message (Case 44353)
- fixed — Transparency slider and image toggle not disabled when only one image and preference for newer content on right (Case 44475)
- fixed — Post-commit reviews show files as reworked, even when they're not. (Case 44493)

© 2003-2011 SmartBear Software

- fixed — Stack overflow when uploading large files. (Case 44497)
- fixed — Local file status is displayed incorrectly with Subversion (Case 44448)
- fixed — Server fails to shut down running on Windows with Java 5 (Case 44633)
- fixed — Tutorial mode preference is ignored in some places (Case 43229)
- fixed — Improve memory footprint and performance of PDF rendering (Case 44519)
- fixed — Uploading URL message is hidden because the dialog is not tall enough. (Case 44661)
- fixed — Fix addversions command
- fixed — Reordering of custom fields can fail (Case 44692)
- fixed — review-xml includes duplicate conversations (Case 44702)
- fixed — Stack overflow in Eclipse plugin with svn:externals (Case 42824)
- removed — Attach URL feature not supported in CodeReviewer (Case 44555)
- removed — Perforce ACL feature not supported in CodeReviewer (Case 44542)
- removed — Groups feature not supported in CodeReviewer. (Case 44553)
- removed — System wide message feature not supported in CodeReviewer. (Case 44557)

v5.0.5002 (beta) — 2009/04/06

*** FIRST RELEASE OF v5.0 BETA! ***

- PDF Overview of New Features: http://smartbear.com/docs/12_Reasons_to_Upgrade_to_5.0.pdf
- HTML Overview of new features: http://smartbear.com/codecollab-new.php
- Full list of new features: http://smartbear.com/codecollab-5.0-all-features.php

8.2.3 Version 4

v4.0.864 — 2010/02/04
- fixed --- List report 181 results improperly cached (Case 49572)
- fixed --- LDAP lockout can occur if PeerReview Complete is used infrequently. (Case 49526)
- fixed --- "Too many open files" error in Content-Cache Diagnostic (Case 49687)

v4.0.863 — 2009/09/1
- added — Support Perforce Eclipse plugin 2009.1.209672 (cmdline wrapper only)
- fixed — Subversion Eclipse plugin support recognize status "Incomplete" (Case 46930)
- fixed — Some modified Vault 390 files not being found (Case 46754)

v4.0.862 — 2009/08/19
- added — Support MKS variants in change package files (case 46212)
- added — SQL Server JDBC 2.0 driver documentation
- fixed — Update documentation for new name of Vault 4.1.x client .jar download
- fixed — Handle empty SVN comments
- fixed — Multiple button upgrade bug wherein multiple users click the "Upgrade Database" button and multiple upgrades are attempted simultaneously
- fixed — Error in Eclipse plugin "fix configuration" action item (case 46164)
- fixed — .vmoptions file extension

v4.0.861 — 2009/07/08
- fixed — Handle MKS change packages in submitted state (case 45631)
- fixed — p4 -Q does not work before *client* version 2005.2
- fixed — Support multi-line environment variables (Case 43002)
- fixed — Crash in ccollab addchangelist new (Case 45684)

© 2003-2011 SmartBear Software

- fixed — Support for svn checkouts at drive roots (Case 45707, Case 45614, Case 45717)
- fixed — Error uploading TFS shelvesets with new files (case 45709)
- fixed — Better error message for Perforce authentication problem (Case 45778)
- fixed — Wrong Vault predecessor for modified files in VSS mode (Case 45839)
- fixed — SVN output interpretation fix (Case 45775)

v4.0.860 — 2009/06/03
- fixed — NPE in ClearCase 322 hashcode (Case 44189)
- fixed — Error parsing Team Foundation 349 fr-CA date format (Case 45373)
- fixed — TFS 349 usernames with leading backslash (Case 45373)

v4.0.859 — 2009/06/01
- added — Better support for higher Perforce 366 security levels (Case 44899)
- added — Support ClearCase 322 7.1 dates (Case 44914)
- added — Support for wildcards in ccollab addfiles 217 (Case 45201)
- fixed — Remove benign log warnings about not able to contact license server (Case 45003)
- fixed — Multiple Vault 390 configurations not working (Case 44642)
- fixed — Custom field views 432 skipped deleted items (Case 44504)
- fixed — Multiple Authors allowed even when Maximum # of Authors 112 is 1 (Case 45107)
- fixed — Clicking 'next' doesn't do anything for Submitted Perforce Changelists on Eclipse 3.4
- fixed — Modified files not found if username case differs from Vault 390 (Case 44642)
- fixed — Handle Perforce plugin authentication when password not saved (Case 44899)

v4.0.858 — 2009/05/06
- added — Support for subversion 1.6
- added — UI to generate new Node ID (Case 44480)
- fixed — In some cases, reordering of custom fields can fail (Case 44692)
- fixed — Stack overflow in eclipse plugin with svn:externals (Case 42824)
- fixed — TFS autodetect causes error when TFS is installed and a file is completely unmanaged
- fixed — Locale sensitive bug in embedded database initialization (Case 44763)
- fixed — Bad VSS login argument on ss.exe command line (Case 44745)
- fixed — Custom report filters for meta-data drop-downs need to show disabled items (Case 44504)
- fixed — Exception when using subversion at a drive root (Case 44961)

v4.0.857 — 2009/04/15
- added — Support for Perforce eclipse plugin 2008.2.195317 (Case 43854)
- fixed — Incorrectly reporting "num-defects" as "num-comments"
- fixed — MKS 358 trunk version limit error (Case 44275)
- fixed — MKS 358 history limited to 200 versions (Case 44275)
- fixed — MKS 358 password appears in debug log (Case 44322)
- fixed — Improper error handling when database fails to initialize (Case 44462)
- fixed — Eclipse plugin support for ClearCase (Case 44561)
- fixed — Only list shelvesets 349 for configured user
- fixed — Broken manual link (Case 44506)

v4.0.856 — 2009/03/30
- added — Ability to upload diffs from a file by name
- fixed — Update documentation on Tomcat auto-deploy for ROOT.xml changes (Case 43973)
- fixed — Updated 'ccollab admin batch' help text
- fixed — Fixed lack of support for multi-lined strings in PHP
- fixed — Exported reports should use UTF-8 character set so as to not garble any characters (Case 41853)
- fixed — Cannot parse certain CVS diffs (Case 44090)

© 2003-2011 SmartBear Software

- fixed — Fix broken admin custom fields icon
- fixed — Missing files in MKS changelist (Case 43812)
- fixed — NPE on uncontrolled CVS file (Case 43962)
- fixed — ClearCase host name incorrect (Case 44007)
- fixed — Cannot create roles or templates (Cases 44172, 44255)
- fixed — Engine.NotificationCreate() method should be public (Case 44276)

v4.0.855 — 2009/03/05
- added — updates to Examples.java
- fixed — Linux client connection problems to MKS Integrity Server (Documentation update) (Case 42389)
- fixed — Split Role setting 'Can change defects' into 'Can change own defects' and 'Can change other user's defects' (Case 43765)
- fixed — Extra space at end of url causes problems in Eclipse plugin (Case 43798)
- fixed — Downloading files from a review with duplicate file names causes a corrupted ZIP file (Case 43796)
- fixed — SourceGear Fortress support in Vault (Cases 42962, 43783)
- fixed — NPE in AccuRev due to "file doesn't exist" (Case 43841)
- fixed — Long changelist description with Unicode characters and Oracle back-end fails to upload (Case 43535)
- fixed — Run each task in a separate thread (Case 43790)
- fixed — ClearCase 'Operation requires a view' exception with supporting documents in review (Cases 43898, 43920)

v4.0.854 — 2009/02/23
- fixed — Perforce ccollabupdatechangelist trigger slow for many thousand files (Case 43688)
- fixed — Search results can return reviews that user cannot access (Case 43722)
- fixed — Unable to parse file extension in ClearCase extended paths, prevents syntax highlighting (Case 43746)

v4.0.853 — 2009/02/11
- fixed — Encode server logs in UTF-8
- fixed — File permission error writing MKS log file under Linux (Case 43241)
- fixed — Javascript bug in classic diff view
- fixed — Bug id markup and HTML encoding conflict (Case 43479)
- fixed — Problem parsing incomplete CVS diffs (Case 42803)
- fixed — Support unmanaged files in MKS directories (Case 43333)
- fixed — Support unmanaged files in TFS directories (Case 43375)
- fixed — Improve chat performance with large numbers of conversations
- fixed — Exception on renamed TFS files when uploading rework (Case 43509)
- fixed — Cannot access view column names in Oracle (Case 43514)

v4.0.852 — 2009/01/16
- fixed — Encode server logs in UTF-8.
- fixed — Do not create duplicate users that vary only base case (effects Oracle and embedded database users) (Case 42741, 42420)
- fixed — Do not add authors to a review if their user account is inactive (Case 43056)
- fixed — If Subversion username is specified, always specify a password (Case 43180)
- fixed — If an XML file includes a charset marker for an unsupported character set, try to recover using autodetection (Case 43182)
- fixed — Add more context to CVS diffs (Case 42803)
- fixed — Oracle limited to 127 switches in a CASE statement
- fixed — Update Vault documentation

© 2003-2011 SmartBear Software

- fixed — Handle no-longer-scheduled-to-be-added and -deleted directories in Subversion (Case 43204)
- fixed — Perforce '-Q' option only supported after 2005.2.
- fixed — New file in ClearCase activities appear as modified (Cases 39166, 43227)

v4.0.851 — 2008/12/16
- fixed — Login should accept 64 characters to match the username database field. (Case 42836)

v4.0.850 — 2008/12/12
- fixed — Perforce changelists with high Unicode characters in filenames or changelist descriptions get text corruption. (Case 41819)
- fixed — Add p4charset global option for interacting with Unicode Perforce servers.
- fixed — Build with Vault 4.1.4 API (Case 42625)
- fixed — Launch browser in daemon thread (Case 42826)

v4.0.849 — 2008/12/03
- fixed — Peak usage by day still broken (Case 42748)
- fixed — Support TFS versions with deletion ids (Case 42331)
- fixed — ScmRevertedLocalCheckout cast error (Case 42450)
- fixed — StarTeam uploads file with multiple files in different subdirectories (Case 42523)
- fixed — Argument replacement bug (Case 42525)
- fixed — Multiple-file external diff on IE 6; cache headers cause file not found (Case 42601)
- fixed — Error when loading transaction from AccuRev without workspace (Case 42604)
- fixed — Unified diffs from Mercurial break diff parser (Case 42500)
- fixed — Allow changing case of login name (Case 42675)
- fixed — P4Win integration pause on error (Case 42677)
- fixed — No way to resolve text when review content is decoded in wrong character set (Case 41819, 42536, 42743)
- fixed — Support Perforce sandboxes on UNC paths (Case 42726)
- fixed — Phone numbers not displayed in tooltips (Case 42707)

v4.0.847 — 2008/11/13
- added — Associate .inl files with C++ syntax highlighting (Case 42188)
- added — Documentation of Active Directory configuration with security groups.
- fixed — Show/hide previously uploaded changelists toggle does not remember state (Case 41246)
- fixed — Better handling of corrupted paswords in config files (Case 42283)
- fixed — Make client commit action get all most recent versions instead of active changelists.
- fixed — Do not display SCM passwords in debug log
- fixed — Duplicate user accounts with leading/trailing spaces (Case 42420)
- fixed — MKS IllegalArgumentException (Case 42252)
- fixed — Add changelist id to review-xml (Case 42520)

v4.0.846 — 2008/10/28
- added — Support for Visual SourceSafe diffs (Case 41339)
- fixed — Database connections dying "randomly" causing intermittent web site failure and stopping the activity-update thread
- fixed — ClearCase server paths are canonicalized
- fixed — Wrong ClearCase predecessor after version list edit (Case 41561)

v4.0.845 — 2008/10/13
- fixed — restrict `ccollab admin review-xml` according to access rules (Case 42080)
- fixed — consistent behavior for ignored and/or unmanaged files in `addchanges` and `addfiles`
- fixed — Make RPM installer cleanup of old files optional (Case 41786)
- fixed — Addfiles fails on ucontrolled files in CVS directory (Case 41590)
- fixed — Disabled users should not receive notifications (Case 41607)

© 2003-2011 SmartBear Software

- fixed — Use correct system admin login name in "license exceeded" error message
- fixed — "Total Person Time" not the same on review summary and review detail (Case 41118)
- fixed — Open home pages will not count towards licensing (Case 41804)
- fixed — Count chat refreshes against licensing only if the user is active.
- fixed — Prepend External Diff path disambiguation instead of appending so file extension is not changed.
- fixed — Config-only dump skipped review templates' custom review and defect field associations (Case 41826)
- fixed — Add Office 2007 file extensions to default binary formats.
- fixed — Error running ccollab set collab command with no argument. (Case 42028)
- fixed — Fix NullPointerException adding uncontrolled files (Case 41988)

v4.0.843 — 2008/09/23

- added — Support for Vault proxy settings (Case 40815)
- added — Scripting 443 section to manual
- added — Mirror Defects to external issue-tracker 445 scripting tutorial to manual
- added — Add checklist to new Review scripting tutorial to manual
- added — Sync Perforce Users 448 scripting tutorial to manual
- added — Time to run report to report information display
- added —Jump next/prev should not wrap without confirming. (Cases 41431, 41594)
- added — Upload comment option for Starteam (and other) add*diffs commands (Case 41745)
- added — MKS changes for subsandboxes
- fixed — NPE in getRecentlyUsedServers with Team Foundation 2008 (Case 41618)
- fixed — Single-click flicker when double-clicking tray notifier
- fixed — Text variable typo
- fixed — Missing TFS files when server and local paths differ in case (Case 41468)
- fixed — Gap in usage graphs (Case 41396)
- fixed — Chat on line 1000000 should be disallowed if possible and promoted back into reasonable range. (Case 41399)
- fixed — Oracle "table or view doesn't exist" bug (Case 41184)
- fixed — Detection of SourceForge ClearCase Eclipse plugin (Case 41646)
- fixed — External Diff launcher error for Perforce Pending Changelists (Case 41515)
- fixed — Compare drop-down needs clarification (Case 41644)
- fixed — Vault takes too long to find modified files (Case 41558)

v4.0.842 — 2008/08/29

- added — ccollab commit prompt for upload comment (Case 41055)
- added — support for GUI Client on Solaris (Case 41144)
- added — allow setting multiple-line custom field values on the command line by getting the field's value from a file. (Case 41176)
- added — 'ccollab admin batch'
- added — Support alternate Team Foundation user names
- added — "Open GUI" menu item to tray notifier (Case 41359)
- added — "support..." email links to GUI Client and Tray Notifier (Case 41343)
- added — "--xpath" and "--xsl" options to 'ccollab admin review-xml'
- fixed — Support alternate Team Foundation user name (Cases 41164, 41196)
- fixed — Fix unified diffs for Lua (Case 41195)
- fixed — role configuration admin screen where drop-down boxes were too wide for the form, causing the page to be extremely wide
- fixed — Clearcase error with Unix paths (Case 41100)
- fixed — make next/prev buttons only stop on differences in current comparison (Case 41001)
- fixed — ability to set custom field drop-down items from command-line (Case 39720)
- fixed — ccollabgui client not installed properly by Linux RPM (Case 41115)

© 2003-2011 SmartBear Software

- fixed — multi-file external diff (Case 41172)
- fixed — Unicode names get abbreviated incorrectly (Case 41150, 41300)
- fixed — Admin users should be allowed to add changelists to any review (Case 41330)
- fixed — ClearCase eclipse plugin integration cannot find view (Case 41302)
- fixed — User comboboxes fail with internationalized names.(Case 41205)

v4.0.841 — 2008/08/20
- added — Command to set custom field dropdowns from the command line (Case 39720)
- added — Support for GUI client on Solaris (Case 41144)
- fixed — Change contact email address for licensing issues to the appropriate Smart Bear email address. (Case 41037)
- fixed — Role configuration admin screen select boxes too wide.
- fixed — Buttons on Linux displayed even though no SCM selected. (Case 41050)
- fixed — Clear Case error with Unix paths. (Case 41100).
- fixed — Server can fail to start if installation path includes a space on Unix.
- fixed — After changing revisions, Next and Prev buttons stop on differences from prior revision (Case 41001)
- fixed — GUI client not installed correctly by Linux RPM (Case 41115)

v4.0.840 — 2008/08/05
- added — Faster Vault integration
- added — More support for MKS integration
- added — Eclipse v3.4 support for automatic upgrade site (Case 40495)
- added — Auto-detect `tf.exe` location for TFS 2008 (Case 40852)
- fixed — Enterprise organization field width too small (Case 40263)
- fixed — Inconsistent letter casing when uploading mixed-case paths from Windows (Case 40170)
- fixed — Notification emails encoded in system default character set can be garbled (Case 40693)
- fixed — Added more keywords to TCL syntax coloring (Case 40620)
- fixed — Vault was prompting for each file instead of for all files together (Case 40619)
- fixed — Proper handling for file renaming in TFS integration (Cases 39946, 40018)
- fixed — Erroneous error for "`tf properties`" (Case 40613)
- fixed — Support for AccuRev integration on OpenJDK (Case 40699)
- fixed — Sundry ClearCase integration issues (Cases 41026, 40198, 40132, 40437, 40619, 39766)
- fixed — ClearCase "unable to access" error when loading from root of Windows dynamic view (Case 41026)
- fixed — Filling output.log with error messages when a session is created and invalidated after response bytes begin (Case 41043)

v4.0.839 — 2008/07/14
- added — Support for Subclipse 1.4 (Case 40457)
- added — ccollab browse --review option 232 accepts "last" and "ask"
- fixed — ccollab adddiffs 219 should put source info in changelist comment (Case 40550)
- fixed — Don't error if there are zero review custom fields (Case 40566)
- fixed — Clarify options for ccollab login 214 (Case 40240)
- fixed — Don't print Subversion password in GUI client (Case 40649)

v4.0.838 — 2008/07/07
- added — Defect permission configuration setting: Don't allow edit/delete defects even if the user was the creator.
- fixed — Upgrade from 2.1.x fail with SQL Server 2005

v4.0.837 — 2008/07/02
- added — Support for Subversion 1.5

© 2003-2011 SmartBear Software

- added — Subcommand 'admin review copy-participants'
- fixed — Added indexes for better query performance (Case 40290)
- fixed — User cannot delete a defect they created (Case 40399)

v4.0.836 — 2008/06/24

- added — File commit support for TFS and ClearCase
- fixed — Better error messages in Perforce triggers
- fixed — Report errors in p4 print
- fixed — Incorrect metrics for "loc changed" in detail report
- fixed — Sort user list based on activity (Case 39030)
- fixed — Some metrics not working when using internal database (Case 40006)
- fixed — When large numbers of users in system, new review screen fails to load (Case 40339)
- fixed — Upgrades from early 4.0 versions fail (Case 40332)

v4.0.835 — 2008/06/17

- added — Add Subversion diffs UI in GUI client 399
- added — Add CVS diffs UI in GUI client 303
- added — Add Perforce diffs UI in GUI client 369
- added — Add StarTeam diffs UI in GUI client 297
- added — Add AccuRev diffs UI in GUI client 289
- added — Role permission 110 to allow userto modify, but not delete, defects (Case 39630)
- added — Vault 4.1 390 support
- added — Syntax highlighting for TCL
- added — JMX monitoring 60 of licensing and users
- added — Peak usage chart 123 of license usage with accurate data
- added — Browse... button to GUI client 189 SCM dialog
- added — Track number of rejected logins due to licensing issues (Cases 39775, 39767)
- added — Integrated support for checking in reviewed materials
- fixed — improved performance of adding ClearCase versions with addversions and addactivity (Case 38732)
- fixed — improved performance of review overview screen (Case 39662)
- fixed — browser integration on OS X (CAse 39690)
- fixed — partial fix for external diff config manifest issues (Case 39752)
- fixed — autodetect tf.exe location (Case 39732)
- fixed — attach materials page missing image (Case 39526)
- fixed — Names with apostrophes and hyphens are not properly abbreviated (Case 39750)
- fixed — Optimize/fix adding versions by name (Case 39281)
- fixed — Do not change the case of mixed-case names when abbreviating
- fixed — No vertical scrollbar in classic view in IE7 (Case 39830)
- fixed — Include external diff button for binary files (Case 39754)
- fixed — User list should not be case sensitive (Case 39869)
- fixed — Performance improvement; less frequent access to assignments table (Case 39590)
- fixed — ccollab addchanges fails on Subversion unmodified file after branch (Case 39880)
- fixed — Bad data causes dump to not restore properly (Case 39623)
- fixed — Team Foundation script output appearing in version content (Case 39887)
- fixed — Links in external defects doubly encoded (Case 39921)
- fixed — Make "private" review field available in reports (Case 39900)
- fixed — Improve performance of default review reports
- fixed — Files missing from diff (Case 40002)
- fixed — Diff view does not reload entire page when options change
- fixed — Diff view preference for wrap lines not honored in single version view (Case 39885)
- fixed — Cannot cancel "Track Externally" by choosing "Edit Defect" instead (Case 39073)
- fixed — Review-only dumps corrupted (Cases 39975, 40060)

© 2003-2011 SmartBear Software

- fixed — Do not log passwords in cleartext on Windows (Case 40010)
- fixed — Firefox 3 popup calendar sized incorrectly (Case 39723)
- fixed — Delay review creation so users do not create spurious reviews (Case 40031)
- fixed — Tomcat logging should be enabled by default at INFO level.
- fixed — Handle "No Data Given" error parsing spurious TFS properties (Case 39946)
- fixed — Exception in Perforce trigger when changelist description is empty (Case 40085)
- fixed — GUI client should explain that it does not support default Perforce changelist (Case 39948)
- fixed — Filenames with characters outside default character set get garbled (Case 40124)
- fixed — 'ccollab addchanges' on deleted filenames (Case 39483)
- fixed — Error parsing Team Foundation dates with DBCS chars (Case 40145)
- fixed — Performance imrpovement when loading review phases (Case 39860)

v4.0.834 — 2008/05/12

- added — Custom report field 178 for number of open defects (Case 39234)
- added — Added 'force-new-browser 205' option to force new browser window
- added — More validation of Browser and Server URL values in GUI Client
- added — Added 'cvs-exe' global option in case CVS executable is not in PATH
- added — Performance improvements to the diff viewer
- added — Added 'accurev-exe' global option in case AccuRev executable is not in PATH
- added — Added 'svn-exe' global option in case Subversion executable is not in PATH
- added — Eclipse plugin Subclipse integration prompt to automatically switch to SVNKit if necessary
- added — Ability to disable reporting 76 in the Web UI (Case 39550)
- added — Better defect usage description
- fixed — Exception in Database Diagnostic (Case 39323)
- fixed — AccuRev NPE on added file (Case 39195)
- fixed — Prevent users from deleting/canceling reviews (Case 38989)
- fixed — Consistent ordering of changelists by SCM in review materials (Case 39159)
- fixed — Eclipse plugin should prompt to save modified files before uploading them (Case 34752)
- fixed — Scm systems not being mapped correctly when uploading changelists
- fixed — Subversion moves (add with history) should show the metrics from the diff (Case 39512)
- fixed — Error when changing default value (Case 38091)
- fixed — Deleted files not included in external diff package (Case 39529)
- fixed — Handle error getting modified files in Add to Review wizard (Case 36685)
- fixed — Perforce (P4V) integration 381 does not report errors (Case 39532)
- fixed — Perforce (P4V) integration 381 does not work when Workspace not specified (Case 39532)
- fixed — Last line of code removed from review if blank (Case 38957)
- fixed — "AccuRev not in working directory" message comes out as "unexpected error"
- fixed — Update user last activity much less frequently (Case 39590)
- fixed — Deadline status should not display when deadlines disabled

v4.0.833 — 2008/04/28

- added — GUI client remembers last four (4) server URL's 194 (Case 38271)
- added — Support for locale-specific TFS dates.
- added — Flag to disallow non-author uploads 77 (Administrator Setting) (Case 38619)
- added — Added "browse" 232 subcommand to clients
- fixed — Subscriptions should not be applied for users whose accounts are disabled (Case 39270)
- fixed — Fix addchangelist help for TFS shelvesets
- fixed — Include summary metrics for all versions in the review detail report. (Case 39228)
- fixed — Add indicator to the "Compare" menu to indicate what is being shown (Case 39219)
- fixed — External diff launcher arguments should be quoted
- fixed — Diff viewer should have a minimum on the number of skipped lines (Case 38119)
- fixed — Add a meaningful error message for unsupported TFS files
- fixed — Fix some bad error messages in client. (Case 39426)

© 2003-2011 SmartBear Software

- fixed — Give a better approximation of disk usage in archiving (Case 39333)
- fixed — Diffs not showing with addsvndiffs
- fixed — ccollab --debug option tells user where log is being saved
- fixed — Subclipse integration doesn't support anonymous repositories (Case 39203)
- fixed — Database dumps are incomplete when dumping a single review (Case 39420)

v4.0.832 — 2008/04/10

- added — Support for AccuRev Eclipse plugin
- fixed — Solaris does not understand test -e (Case 39079)
- fixed — Diff uploads appear to pick incorrect version (Case 38974)
- fixed — Defect description on review overview omits anything that looks like a tag (Case 38708)
- fixed — Notification emails stop working after server restart (Case 39102, 39066, 39173)
- fixed — Case error in custom field documentation (Case 39197)
- fixed — Performance improvement generating diffs (Case 39079)

v4.0.831 — 2008/04/07

- added — User creation trigger 115 (Case 38762)
- added — Allow users to set skip unchanged 131 option in diff viewer (Case 39051)
- added — Option to disallow non-author uploads 77 (Case 38619)
- added — `ccollab set` 215 prompts if no option value specified (Case 36979)
- added — `ccollab set` 215 displays value for all options if no option specified (Case 36724)
- added — Support GIT diff variant with `ccollab adddiffs` 219 (Case 39048)
- added — External diff preset 280 for Beyond Compare (Case 38988, 37588)
- added — PeerReview Complete GUI Client support for Linux x86_64
- added — Syntax 161 highlighting for SQL (Case 38300)
- added — Tooltip showing when reworked files 152 were last updated
- fixed — Correctly set windowing system parameter in Linux RPM client installer
- fixed — PeerReview Complete GUI Client should persist size and location (Case 38906)
- fixed — Improve progress messages for `ccollab addchanges` 216 (Case 36423)
- fixed — IE "remember me" cookies (Case 38947)
- fixed — syntax highlight "sbyte" as a keyword in C#
- fixed — Update P4V Tools import file for P4V 2007.3 (Case 38994)
- fixed — Syntax highlighting for PHP (Case 36489)
- fixed — When there are too many users, review creation page is slow (Case 38688)
- fixed — Better support for nested Subversion working copies (Case 38979)
- fixed — Only administrators can run `ccollab admin syncusers` 235
- fixed — Update SQL Server drivers; drop support for SQL Server 7.0
- fixed — `ccollab adddiffs` 219 fails if files have different names (Case 37940)
- fixed — Minor performance improvements in diff emitting code
- fixed — Don't display time in review deadline (Case 38948)
- fixed — Bad links in server upgrade section
- fixed — `ccollab addfiles` 217 doesn't pick up SCM configuration (Case 36538)
- fixed — Review not displaying files (Case 39093)

v4.0.829 — 2008/03/26

- added — Support for GIT-style diffs in `ccollab adddiffs` 219
- added — Detect XML "encoding" header attribute and use to parse XML files with correct encoding (Case 38933)
- added — Indicate the number of reworks a file has gone through in the "Status" column of the review summary page 152 instead of just saying "Reworked." (Case 38875)
- fixed — Error trying to upload or view files not in UTF-8 or certain binary files (Cases 38925, 38913, 38933, 38941)

© 2003-2011 SmartBear Software

- fixed — "Log user off" link should be present only if the account is logged in according to floating-seat rules (e.g. auto-log-off after 1 hour), not by whether the user has logged off manually.
- fixed — "New comment" form should be disabled when no line is currently selected
- fixed — ClearCase Windows version paths garbled in viewer drop-down

v4.0.828 — 2008/03/25

- added — Separate licensing button "update from smart bear" 44 to reduce confusion
- added — System property to specify the cache sizes for line parse and syntax coloring cache
- added — JMX beans for cache sizes; framework for other JMX beans
- added — More review information in notification emails (Case 38618)
- added — Include server configuration files in debugging dump 70
- added — Debug and migration dumps 70 use form to select options instead of list of links
- added — Option to include server logs in system data dumps 70
- added — Incorporate review dumps 71 into the new data dump form
- added — Team Foundation Server 349 support (beta)
- added — Add "log this user off" 86 to administrative user list (Case 38845)
- fixed — improved performance of caches for syntax and diff highlighting.
- fixed — ccollab addchanges 216 should not create review when no files are selected (Case 36649)
- fixed — Error on some JVMs running addcvsdiffs 221 (Case 38723)
- fixed — Getting blank screen when trying to input new license code (Case 38830)
- fixed — Binary characters blocking chat from working (Case 38825)
- fixed — User Login Prompt field and others should highlight hyperlinks automatically (Case 38841)
- fixed — "Log off" from one browser should log out of all browsers
- fixed — Password reset does not reset password for admin user (Case 38825)
- fixed — Improve performance of scm output processing (Case 38864)
- fixed — Force clear of PeerReview Complete 2.1 cookies

v4.0.825 — 2008/03/18

- added — Fun facts 83 help
- added — Client 'admin review finish' 243 command
- added — Users can select browser 205 to launch (Case 36294)
- fixed — Button text in planning phase of Review Overview page (Case 36946)
- fixed — Restore links to metrics definitions topic in manual
- fixed — Refreshing action items from clients counts you as logged in (Case 38546)
- fixed — Roles without participants should not be empty in table (Case 37001)
- fixed — Confusing instructions in the defect log box (Case 37002)
- fixed — AutoDetectingReader not working properly (Case 38239)
- fixed — Initialize database button style (Cased 37080)
- fixed — Rework cookie handling to comply more completely with Tomcat's specifications
- fixed — Add more information on Perforce trigger errors (Case 38624)
- fixed — Defects showing up with comment icons in the diff view gutter. (Case 38502, Case 38651)
- fixed — Client 'addsvndiffs' command fails with Cygwin svn. (Case 38285, Case 37943)
- fixed — Client installer reports server is too old when the server is unavailable. (Case 38682)
- fixed — Client prints stack trace when adddiffs finds no diffs (Case 38547)
- fixed — Update Tomcat jar to fix LDAP authentication issues (Case 38795)

v4.0.824 — 2008/03/06

- added — User configurable Fun Facts 83 (Case 37885)
- added — Ability to get more than 10 search results 174 (Case 38454)
- fixed — Action items need to update immediately when the tray notifier is clicked.
- fixed — Trying to auto-detect AccuRev and throwing exception instead of just skipping it. (Case 38197)
- fixed — Defects by User metrics differ from defects list report (partial fix). (Case 38443)

© 2003-2011 SmartBear Software

- fixed — Reports throw exception when filtered on invalid date (Case 38500)
- fixed — Use a different browser launching utility.
- fixed — Auto-login cookies are not cleared in IE after an upgrade from 821 (or earlier) to 823 (Case 38592)
- fixed — Various text updates for UI consistency (Case 37060)
- fixed — GUI clients should not log passwords in plain text.

v4.0.823 — 2008/03/03

- added — Select suggested new review title (Case 37152)
- added — Cache action items 139 refresh connection for 20 mins
- added — Ability to capture debug log 196 in GUI Client
- added — Upgrade Tomcat to 5.5.26
- fixed — Handling of system-administrator in LDAP configuration
- fixed — File upload description is not unicode clean (Case 36875)
- fixed — Clients ignoring proxy settings
- fixed — Change default action items refresh interval to 5 minutes
- fixed — Widget is disposed in tray notifier (Case 38262)
- fixed — Double quotes in chat turn in to quadruple quotes (Case 38408)
- fixed — Large text field doesn't resize automatically on "edit review" screen (Case 38405)
- fixed — Strike-through on defect text should not include "tracked external as" portion (Case 38208)
- fixed — Recognize CVS for individual files (Case 38417)
- fixed — Inserts or deletions at the end of file prevent classic view from loading (Case 37984)
- fixed — Custom fields editor shows wrong inputs when validation fails (Case 38214)
- fixed — Metrics discrepancies (Case 38443)
- fixed — Embedded database search results are case-sensitive (Case 38473)
- fixed — Defect report label is "Severity" when it should be "Type" (Case 38486)
- fixed — Reinstate user preference for disabling syntax highlighting
- fixed — Safari & Opera render <wbr> tag incorrectly (Case 38488)
- fixed — GUI Client and Tray Notifier should write to different log files (Case 38378)
- fixed — Keyword highlighting within intra-line diffs

v4.0.821 — 2008/02/21

- added — When word-breaking long continuous words, try to break on camelCase boundaries if possible (Case 38289)
- added — SCM server information now in the `<artifacts>` section of "`ccollab admin review-xml`" (Case 38261)
- fixed — Links in chat are corrupted, both hyperlinks and bug-links (Case 38319, 38337, 38330)
- fixed — Debug review dump cannot be unzipped (Case 38036)
- fixed — Custom report fails to run when "Idle Time" column is selected under SQL Server (Case 38240)
- fixed — Script error on review overview when review is in the completed phase
- fixed — Erroneous error "version content already sent" when using diff-shim application with Subversion under Windows (Case 38225)
- fixed — Tray notifier displays erroneous error (Case 36402, 37744, 36237)
- fixed — Erroneous "`<wbr>`" text displayed inside URL's (Case 38195, 38215)
- fixed — Make label for "Local path" more clear in the cross-platform GUI client (Case 38066)
- fixed — Fixed incorrect comment label of "`DEDT`" on upgraded databases (Case 38336)
- fixed — Custom review filter is partially ignored when looking at printable format (Case 38269)
- fixed — Incorrect Subversion check-in comment parsing (Case 38285)
- fixed — Hide and encrypt password configurations in the cross-platform GUI client (Case 37830)
- fixed — Large files were sometimes not showing up in the diff viewer (Case 38235)

v4.0.820 — 2008/02/12

© 2003-2011 SmartBear Software

- added — Support for sym-links 397 in Subversion (Case 38038 and Case 38039)
- added — Reports and views 179 now count "number of comments" as the number of comments made by users in the context of chatting, not including system messages like "created defect" or "file uploaded." (Case 37645)
- fixed — Bug upgrading older databases to build 819 (Case 38189)

v4.0.819 — 2008/02/07

- added — Log off link 125 on the menu bar. (Case 37923)
- added — Update button 44 on licensing screen to force update of license codes from SmartBear servers.
- added — Support for Perforce Eclipse Plugin (P4WSAD) version 2007.3.601 (Case 37916)
- added — Installer option to preserve existing database settings. 34
- added — New icons for various operations
- added — Chat icons in diff window 157 gutter show unread conversations 169 with unread conversation icon.
- fixed — Oracle strings now support more than 1023 characters
- fixed — Vault integration case sensitivity.
- fixed — After completing a review to move it to rework, a refresh moves the review back to inspecting (Case 37927)
- fixed — Review cancel action should take user back to home page. (Case 37938)
- fixed — Adding files by diff does not handle rework status correctly (Case 37902)
- fixed — Support version name /main/0 as a valid predecessor
- fixed — Reduce chat load times.
- fixed — Template selection alignment is inconsistent (Case 38004)
- fixed — Support Subversion in languages other than English (Case 37537)
- fixed — Use "last changed revision" as Subversion local version
- fixed — AccuRev failing on Linux with capital letter in path (Case 37932)
- fixed — Asynchronously load file content for better browser performance.
- fixed — Trailing context lines has one too few lines. (Case 38020)
- fixed — Do not send multiple copies of emails to the administrator (Case 37954)
- fixed — Long text on review overview makes some information flow offscreen.(Case 37129)
- fixed — Links for dumping system data are confusing (Case 37922)
- fixed — AccuRev cannot auto-detect from path (Case 38052)
- fixed — When rolled up and changelist comments repeat, save space by indicating the number of repetitions
- fixed — Misaligned diffs (Case 38026)
- fixed — Restore ability to change the login prompt
- dropped — Eclipse plugin no longer supports Eclipse 3.0 (Eclipse 3.1 or better is required)

v4.0.818 — 2008/01/24

- added — Added usage statistics 123 sub-page to the Admin section
- added — Added documentation 189 for the GUI Client
- added — Added documentation 278 for the Tray Notifier
- added — Added documentation for `ccollab login` 214 command
- added — Downloadable Eclipse plugin as zipped-up eclipse update site (Case 36063)
- fixed — Log version number in GUI Client log
- fixed — Moving the attach file dialog on the review edit screen causes the dialog to disappear. (Case 37658)
- fixed — Restore ability to edit title after review completion (Case 34178)
- fixed — Database unavailable at server startup makes PeerReview Complete require a restart. (Case 37848)
- fixed — Add*Diffs 200 commands use of GUID's for changelist id is confusing to users (Case 37640)
- fixed — IE 7 does not like RSS feeds with DTD's in them (Case 37890)

© 2003-2011 SmartBear Software

- fixed — Overdue notice showing up on complete reviews on the action items lists (Case 37587)
- fixed — Invalid Subclipse info for svn servers with no path (hostname only) (Case 37665)

v4.0.817 — 2008/01/18

- fixed — Incorrect diffs from Subversion (Case 37133)
- fixed — Print product version number in Eclipse log (Case 37648)
- fixed — Script errors in classic view
- fixed — User short names are duplicated if names are too similar (Case 37704)
- fixed — Client installer no longer kills running client taskbar app (Case 37314)
- fixed — Predecessor version not found if file has many version; ClearCase only (Case 37666)
- fixed — Cannot configure multiple Perforce servers (Case 37736)
- fixed — ContentViewer must havea content provider when input is set (Case 37755)
- fixed — ASCII control characters in files cause "More Lines" operation to hang (Case 37718)
- fixed — GUI client needs to allow SCM specification and local path
- fixed — addactivity subcommand should show local synced version numbers (Case 37364)
- fixed — IE 6 fails to download custom reports
- fixed — Added ccollab logiin subcommand (Cases 36731, 36729, 36730)
- fixed — Comments misaligned at the point of a code insertion (Case 37747)
- fixed — Mandatory subscriptions not uniformly enforced on the server
- fixed — Clear Case fixes for named local versions
- fixed — CVS uploads from GUI client do not work (Case 37800)
- fixed — Make action item url's work in non-root contexts
- fixed — User account creation fails from administration screens (Case 37823)
- fixed — Grace seats are not handled properly

v4.0.814 — 2008/01/08

- added — Added commands "ccollab admin review participant assign 245", "ccollab admin review participant remove 246", and "ccollab admin review comment create 247"
- added — Added command "ccollab admin review defect create 249"
- added — Global option for SMARTBEAR_PROCESS_USER_WAIT
- added — Support clients working in ClearCase view directory (Case 37082)
- added — --creator option to "ccollab admin review create 240"
- added — add "ccollab admin review defect mark-external 252" and support defect id "last 252"
- added — trigger talkback 115 to set review access restriction (Case 36783)
- fixed — New user registration on login page fails (Case 37448)
- fixed — AccuRev integration predecessor algorithm should be "previous occupant" (Case 36970)
- fixed — Make skip unchanged preference work with new diff viewer (Case 36422, Case 37397)
- fixed — Intraline diff highlight expands to include entire SGML tags, making it hard to understand what actually changed. (Case 35430)
- fixed — 'ccollab set username' has poor error messaging (Case 37225)
- fixed — ' ccollab info' messaging has bad formatting (Case 37226)
- fixed — adddiffs with no third argument causes NPE (Case 37303)
- fixed — Default Perforce p4port to 1666 if not specified anywhere at all
- fixed — auto-detect ClearCase in view root directory
- fixed — Custom reports creates links too long for IE (Case 37551)
- fixed — Diff viewer title should lead with file name (but not full path), so it will show up nicely in the task bar. (Case 37547)
- fixed — Underscores occluded when intraline diffs wrap (Case 37436)
- fixed — Error in toolbar application when launching application twice (Case 37550)
- fixed — Empty Boolean type options cause NullPointerException (Case 37508)
- fixed — Diff viewer should scroll new selections somewhere toward the middle of the page. (Case 36634)
- fixed — Completed reviews should not use current date to determine if review is "overdue". (Case

© 2003-2011 SmartBear Software

37587)
- fixed — NullPointerException in CvsClientConfiguration. (Case 37574)
- fixed — read config files and ccollabgui scm config settings on a best-effort basis

v4.0.812 — 2007/12/21
- added — Added commands "ccollab admin review create 240", "ccollab admin review edit 241", "ccollab admin review delete 242"
- added — Ability for Administrators to change access rules 76 on reviews in progress (Case 37190)
- added — Friendlier stalled review notifications 134 (Case 37391)
- fixed — Vestigial new review remains after client error (Cases 37393, 23390)
- fixed — `ccollab addchanges` 293 for AccuRev uploading files in lowercase
- fixed — Full line differences not highlighted (Case 37413)
- fixed — Track externally fails for defects in overall section of files (Case 37319)
- fixed — Side by side view has issues displaying deleted files correctly (Case 37412)
- fixed — Error installing server with Oracle or SQL Server databases

v4.0.811 — 2007/12/14
- added — Email notifications 134 contain a footer indicating that the email was automatically generated.
- fixed — Improved performance of diff viewer for large files, especially with Unix or Macintosh line endings.
- fixed — Command line client erroneously reports `ccollab addchanges` works with CMVC.
- fixed — Emails going out from the system administrator rather than the default address (Case 37177)
- fixed — Perforce integration ignoring p4client or p4user if only one is set
- fixed — Better help for the command line client
- fixed — Triggers should not be required to respond with a well-formed document if they're not talking back (Case 37307)
- fixed — AccuRev integration throws NullPointerException in `ccollab addstream`.
- fixed — AccuRev integration displays wrong error message when not authenticated.
- fixed — Subscription fixes and access restrictions (Case 36793)
- fixed — Email notification option to disable for administrators; restoring migration files restores with notifications disabled. (Case 37285)
- fixed — Highlighted text does not honor user preference fonts (Case 37369)
- fixed — After changing the review deadline, the "Apply" button reverts the deadline to the default (Case 37382)

v4.0.810 — 2007/12/7
- fixed — StarTeam differences support in `ccollab addstdiffs` 225 command (Case 37034)
- fixed — ClearCase `ccollab addactivity` 229 should give empty predecessor for files new to activity (Case 37142)
- fixed — Support for ClearCase \0 versions (Case 36675)
- fixed — Error when getting SQL or CSV from custom reports in Internet Explorer (Case 37185)
- fixed — Mark external (defect) fail on review overview page (Case 37274)
- fixed — Tray notifier throws NullPointerException when double clicked upon startup (Case 36879)
- fixed — Perforce integration not including changelist number in review title
- fixed — Perforce integration not copying changelist description into overview field.
- fixed — NullPointerException in `addp4diffs` 224
- fixed — Additional logging in the email notification processor.

v4.0.809 — 2007/12/4
- added — ClearCase performance changes (Case 36819).
- added — Chat pane 158 is masked when loading
- added — Add "review completion date" to `review_activity_summary` 432 view (Case 37196)
- fixed — External diff launcher fails because of Windows absolute file names (Case 37175)

© 2003-2011 SmartBear Software

- fixed — Improved remember me cookie system; works with container managed role-based security.
- fixed — Custom fields copied from previous reviews gets wrong previous reviews (Case 37183)
- fixed — Installer fails to migrate from 2.1 if application is running at a non-root context path.
- fixed — Fix `addversions` 227 argument parsing.
- fixed — Defect tracking integration not showing up for custom fields (Case 37147)
- fixed — AccuRev integration `addstream` 231 was picking incorrect previous versions in some cases (Case 36970)
- fixed — New diff viewer ignores user font preference (Case 37159)
- removed — Undocumented `report` command; would not work with improved security model.

v4.0.808 — 2007/11/28

- added — Added SCM configuration options to cross platform GUI.
- fixed — ClassCastException in `addstdiffs` command (Case 37034)
- fixed — Manual pages were not loading properly in Eclipse
- fixed — Attach changelist fails in IE6 (Case 36932)
- fixed — Clean up font sizes in cross platform GUI for GTK systems.
- fixed — Removed spurious logging when database is known to be invalid.
- fixed — Floating license timeout is now one hour instead of four.
- fixed — Installer fails to remove old jars when updating the Eclipse client (Case 37009)
- fixed — Under certain circumstances, servers running with the embedded database could not be upgraded to 806 or 807 builds.
- fixed — Improve performance of the chat portion of the diff viewer (Case 37127)

v4.0.807 — 2007/11/19

- added — Do not allow reviews to move to inspection phase without materials 146
- added — (AccuRev only) Added support for ccollab addardiffs 220 for uploading arbitrary AccuRev differences with full context.
- fixed — Super-search box didn't handle characters outside the Latin-1 range
- fixed — Action item "Response to Comments" not posted when new changelist contains only updated files (Case 36847)
- fixed — "Compare" menu in diff viewer lists all uploads as "1st" (Case 36931)
- fixed — `ccollab addsvndiffs` now includes full diff context instead of just 3 lines surrounding each change
- fixed — `ccollab addsvndiffs` now supports binary files correctly regardless of local line ending settings
- fixed — Inconsistent table cell alignment on review overview page (Case 36945)
- fixed — Custom fields cannot be reordered (one-off special database export error case)

v4.0.806 — 2007/11/15

- added — Defect state icons in the diff viewer 157
- added — Client commands for administering author subscriptions 258
- added — Client commands for administering file-based subscriptions 260
- added — Home Page menu option in the tray icon context menu 279 (Case 36856)
- added — Web user interface accepts and displays Unicode input (Case 36697)
- added — Add submitted changelists support to cross-platform GUI 189
- fixed — Diff viewer should start with the first diff if user clicks on file name on review overview page (Case 36637)
- fixed — Participant list expansion does not happen on first apply (Case 36632)
- fixed — The `ccollab addstream` command requires being in a local workspace directory (Case 36350)
- fixed — Unable to edit defects in the Overall section (Case 36806)
- fixed — Editing defects does not set custom field drop-downs in Firefox (Case 36807)
- fixed — Under certain scenarios, comments needed to be marked read in all versions of a file before closing a review (Cases 36845, 36572)

© 2003-2011 SmartBear Software

- fixed — Printable report double-html-encoded text (Case 36877, 35540)
- fixed — Custom report "Recently Completed Reviews" did not properly filter against "completed" status
- fixed — Review title not set by default when uploading changelists with descriptive text (Case 36437)
- fixed — `NullPointerException` in tray notifier
- fixed — Enterprise organization field is too small (Case 36904)
- fixed — File subscriptions not working for Perforce paths (Case 36870)
- fixed — Anonymous Subversion has log entries with no author
- fixed — Archiving fails with blank screen. (Case 36973)
- fixed — Improved performance of archive queries.
- fixed — Trivial Reviews report should not contain canceled reviews (Case 36873)
- fixed — File subscription input fields too short; expanded to 255 characters (Case 36939)
- fixed — Explain to the user why the recent participants list is empty (Case 36941)
- fixed — Password overrides and `ccollab set collab` had confusing behavior.

v4.0.805 — 2007/11/5

- added — AccuRev 288 only: New `ccollab addstream` 231 command uploads differences pending promotion given an AccuRev stream name (Case 36530)
- added — User RSS feeds 135 are now identified by a guid rather than by user name.
- added — Review completion date in review overview 149 (Case 36679)
- added — Eclipse review wizard suggests a new review title.
- fixed — Eclipse synchronize view actions missing icons
- fixed — Text   appearing in file overview display (Case 36704)
- fixed — Eclipse "add to new review" and "add to existing review" should be consolidated.
- fixed — Variables not being substituted in stalled review emails (Case 36703)
- fixed — No password (unset) causes issues in Eclipse plugin and system try notifier (Case 36689)
- fixed — ccollab set collab "" does not overwrite server-url properly (Case 36725)
- fixed — Some database dump files could not be reloaded because of improperly encoded special characters (Case 36773)
- fixed — Installer does not properly add strictAuthOnly attribute to realm definition.
- fixed — 4.0 clients fail to authenticate when server is LDAP authenticated. (Case 36739)
- fixed — Do not specify password argument to P4 commands unless specifically configured to do so.
- fixed — Support spaces in subversion committed file names.

v4.0.804 — 2007/10/29

- added — Performance improvements on Review Overview screen
- added — Handle Perforce ticket mode, automatically login for a new ticket if necessary
- added — Use Add to Review Wizard in GUI Client
- fixed — Add view for custom field drop-down values (Case 36607)
- fixed — Bogus scrollbar when window shrinks in non-wrapped side by side diff (Firefox only) (Case 36209)
- fixed — Clicking on skipped lines section puts non-number in the new chat line number (Case 36561)
- fixed — With short filenames in the same directory, icon column is too wide.
- fixed — Binary file type field should be longer than 255 chars (Case 36640)
- fixed — Remove mandatory dependency on Eclipse CVS plug-in (should be an optional dependency)
- fixed — New subscription fails (Case 36656)
- fixed — Improved logging and error messaging when send email fails (Case 36641)
- fixed — Next/Previous highlights the wrong line for deletes (Case 36635)
- fixed — Support spaces in file/path names in Subversion (Case 36360)

v4.0.803 — 2007/10/18

- added — Action items 139 on homepage refresh without a complete page refresh.
- added — Documentation on configuring LDAPS 50

© 2003-2011 SmartBear Software

- fixed — Custom reports broken when using embedded database (Case 36470)
- fixed — addactivity command fails when file paths contain spaces (Case 36463)
- fixed — NullPointerException in ccollab addchanges (Case 36507)
- fixed — Metrics by Review report only showing reviews with defects (Case 36481)
- fixed — Eclipse plugin stores passwords in plain text (Case 35694)
- fixed — actionitems command does not list action items (Case 36505)
- fixed — addchanges command should not be available when SCM is Perforce
- fixed — Oracle backend should store integers as Number(10) so external tools will recognize as Number.
- fixed — Possible fix for tray notifier crashing
- fixed — Comment promotion prevents bad comments from being marked read
- fixed — External diff viewer link should not be displayed when showing only one version (Case 36451)
- fixed — Jump to next line sets the line to -1 if no more defects
- fixed — Installer fails to preserve database credentials on upgrade (Case 36540)

v4.0.802 — 2007/10/12

- added — New field in review_activity_summary view, author_rework_hours (Case 35473)
- fixed — Minor UI improvements on review overview page.
- fixed — Optimized syntax coloring for very long lines and large files.
- fixed — Search box now jumps directly to review if the search text matches a review id. (Case 36390)
- fixed — Selection causes text to move around in some parts of diff viewer. (Case 36210)
- fixed — Firefox refreshes the review overview page twice instead of the once required. (Case 35882)
- fixed — Tray notifier should to to "normal" mode if no more urgent action items.
- fixed — Version ordering on review overview screen was incorrect if user preference set to alphabetic.
- fixed — System tray notifier was not shutting down on upgrade.
- fixed — Upgrading a server with existing LDAP configuration results in 403 errors from the server.
- fixed — In diff viewer, selecting text results in a script error (IE only)
- fixed — When license code is invalid, the node id is not displayed, but is required in order to resolve the issue.
- fixed — Minor changes to documentation.

v4.0.801 — 2007/10/08 **FIRST RELEASE OF v4.0 BETA**

- **BIG NEW FEATURES**
- added — Revamped diff viewer 157 with over-under view 157, much faster loading times, more jump/search features 159, and hide-able chat pane 167
- added — Subscriptions 133 allow users to get on a review with author- or file-based rules
- added — "Recently Completed Reviews" 139 list on home page
- added — Customized Review Report 176
- added — "New Review" 140 page is now a single page rather than a wizard, cutting down drastically on the time it takes to create and start a new review
- added — Open differences in local diff viewer 157
- added — Review deadlines 79
- added — Proactive notification system 134 alerts you to reviews that are stalled
- added — RSS feed 135 for Action Items 124
- added — Command-line structure reorganized and many new options 211 added
- added — Extensive Command-line help 213
- added — Tray notifier 278 for Windows and Linux
- added — Enterprise organization 88 specification for reviews
- added — Annotate files 152 in overview screen with one-line comments that are visible without opening files
- added — SCM system can usually be detected automatically

© 2003-2011 SmartBear Software

- **LITTLE NEW FEATURES**
- added — "Fun Facts 83" feature on home page
- added — User list drop-down 143 now support substring-based searching for names
- added — Custom reporting view `defects_by_file` 432
- added — Command-line client now connects to server much more quickly
- added — Addchanges 216 from multiple SCM configurations in one command
- added — All global options can be overridden 198 on the command-line
- added — Special `last` 242 keyword can be used on the command-line to refer to the review that was last created
- added — Version control system is automatically detected by the command-line client in most cases
- added — Ability to disable all metrics 76 displays (Case 25888)
- added — Ability to edit list of files before they get uploaded for `ccollab addactivity` 229 and ccollab addtrack 230 (Case 35009, 35383)
- added — Home page action items 139 have more information and are split by incoming, outgoing, and more
- added — Many UI elements can now expand and contract
- added — Date input boxes now use a proper calendar widget
- added — Database diagnostics 123 help you and tech support diagnose problems
- added — Cleaned up user activity statistics 84 on the Admin/User page so it's easier to see how many licenses you really need
- added — User initials algorithm now supports "Last, First" semantics
- added — Button on New Review page lets you pick up 141 custom field settings from the previous review
- added — Added on-wire compression for more web page elements for faster page-load times
- added — Speed improvements for the Review Overview page 148, eliminating 60% of the SQL queries
- added — Speed improvements for the Chat pane 167, making the common no-op case fast
- added — Review columns 152 are hard to differentiate with many reviewers (Case 35450)
- added — Improved defect icons 170 for red-green colorblind users (Case 35023)

- **FIXES**
- fixed — Syntax highlighting got confused with certain sequences of escape characters in C-style strings
- fixed — Subversion checkout at root of local disk fails to upload files properly (Case 34992, 35056)
- fixed — New files/directories caused Eclipse plug-in to fail to find other modifications (Case 35664)
- fixed — Users could re-open review by continuing to comment; now must explicitly re-open 149 the review with a button.
- fixed — When server is awaiting database upgrade, clients were reporting "incompatible server" instead of "needs database upgrade" as the error message (Case 32710)
- fixed — Incorrect handling of non-UTF-8 characters in communications with the Perforce server (Case 36367)
- fixed — Reporting database view 432 `review_activity` now includes all participants, not just those with non-zero activity time

8.2.4 Version 2

v2.1.731 — 11/29/07
- fixed — Fix activity changelist editing of LOCAL versions. (Case 36839)

v2.1.730 — 10/30/07
- added — Add `loc_unversioned` field to the `review_version_summary` containing the line count from files not under version control (Case 36613)
- fixed — Improper handling of Unicode characters in Perforce changelist text. (Case 36737)

© 2003-2011 SmartBear Software

- fixed — Subversion integration fails when spaces in file names

v2.1.729 — 10/16/07
- fixed — Installer can throw NullPointerException when installing server under Java 1.4 (Case 36080, 36093, 36247, 36242, 36243)
- fixed — Metrics by review report only shows reviews with defects (Case 36481)

v2.1.728 — 09/14/07
- added — Support for editing the list of files being uploaded for `ccollab addactivity` 229 (Case 35009, 35383)
- fixed — Last activity date was being updated from Windows GUI Client even if the user wasn't logged in, only when a non-floating license code was installed
- fixed — Perforce trigger was blocking check-in on integrate even with `--ignoreintegrate` is set, if additional files were added as part of the integration and therefore tagged as "branch" by Perforce (Case 35897)
- fixed — Now impossible to change login or password if LDAP/Active Directory is being used (Case 36052)
- fixed — User permissions on views were being destroyed when views were recreated in Oracle and MySQL

v2.1.727 — 08/24/07
- added — First release of AccuRev support 288
- added — Customizable text 75 on the login screen instructing the user which account to use (Case 35653)
- added — Recent participant list 143 should go back only 30 days so older users disappear (Case 35685)
- added — `ccollab addchangelist` 218 now supports multiple changelists on the same command-line
- fixed — URL encoding error externalizing a defect when a quote character appeared in the defect text (Case 35638)
- fixed — Comment promotion error with comments on deleted files in rolled-up view (Case 35698)
- fixed — Some custom reports views were not available under Oracle when custom field titles were longer than 30 characters (Case 35420)
- fixed — User preference for "start with latest" versus "start with base" not being honored with mixture of local and committed changelists (Case 35597)
- fixed — Not properly encoding control characters (below 0x20 ASCII) in database dump XML files
- fixed — Installing server component multiple times switching databases between embedded and SQL Server causes invalid configuration (Case 35488)

v2.1.725 — 08/01/07
- added — Variable substitutions 119 for review and defect custom fields 104 (Cases 35552, 35556)
- added — Now only administrators can un-cancel a canceled review (Case 35502)
- fixed — `Next/Prev` buttons broken in the New Review Wizard 140 (regression from build 724) (Case 35491)
- fixed — Default values for defect custom fields not showing up under IE 6 in the side-by-side view chat 168 area (Case 35551)
- fixed — User could download file content from the `/data` servlet without login credentials (Case 35580)
- fixed — User preference 127 for whether to display "previous" or "base" version by default in side-by-side view was selecting the wrong thing with files not under version control (Case 35476)
- fixed — ccollab addchangelist 408 with Subversion 396 puts the wrong text in the title of the review if the changelist description starts with the letter "r" (Case 35452)
- fixed — `ccollab addchanges` 329 with ClearCase 322 invoked from different subdirectories can

© 2003-2011 SmartBear Software

insert "." path components thereby making comments not promote across versions properly (Case 35575)
- fixed — Server installer was not setting the LDAP system administrator property (Case 35477)

v2.1.724 — 07/23/07
- added — Support for Unicode text files in the diff viewer 157
- added — Multiple conversations for binary files in side-by-side view
- added — Support for Python and Visual Basic syntax coloring in the side-by-side view (Case 35427)
- added — Support for the UltraCompare textual diff file format 201 (Case 35343)
- added — Subversion trigger for creating a review 410 should allow for setting the review title and overview text (Case 35252)
- fixed — Double-clicking the "Accept" button when there's text in the comment box can result in duplication of the comment and "Accept" text in the conversation history (Case 35384)
- fixed — `ccollab addcvsdiffs` 221 doesn't use absolute RCS server paths when a file has been added (Case 35345)
- fixed — Review title limited to 128 characters when editing review information after the review has started, whereas the limit is 255 when creating a new review (Case 35399)
- fixed — External bug URL 80 link was not honoring `BUGSUBJECT` or `BUGID` special fields (Case 35419)
- fixed — Files now sorted case-insensitive (Case 34511)

v2.1.723 — 07/13/07
- added — New command-line command `ccollab addfiles` 217 for uploading files not under version control (Case 35268)
- added — Display the official, permanent link to a review at the top of the Review Summary 148 page (Case 35277)
- added -- Now supports Eclipse v3.3; changes include: Removed hidden popup that was making help content invisible, Removed byte order mark in Eclipse help
- fixed — When role disallows editing/deleting defects, participant cannot even edit/delete his own defects (Case 35272)
- fixed — Able to make a new conversation 168 on line 999999 (Case 35317)
- fixed — Spurious error message when uploading files using v2.1 client 196 against v2.0 server (Case 35263)
- fixed — Create-Review trigger 115 was not being executed when the command-line client created the review (Case 35128)
- fixed — Removed hidden popup in Eclipse v3.3 that was making help content invisible

v2.1.721 — 07/05/07
- added — ClearCase `ccollab addactivity` 229 should diff re-bases and integration branch with `--diffintegration` (Case 34382)
- added — Configurable minimum timeout for AJAX chat refresh rate 75, plus restrict per-user refresh rate so as to not swamp the server or the browser with chat-refresh requests
- fixed — Cannot continue the review due to "unread comments" when in fact all comments are read and the "unread" ones are just rolled up (Case 35163, 35104)
- fixed — Some trigger variables 119 were not being replaced correctly (Case 35128)
- fixed — Client installer 182 doesn't check for an existing trailing semi-colon when updating the Windows `PATH` environment variable (Case 34989)
- fixed — Stop logging spurious error messages when the database has no tables (Case 34890)
- fixed — ClearCase version of `ccollab addversions` 227 not properly handling files with predecessor version `main/0` (Case 35144)
- fixed — Can't get past "database needs updating" screen when MySQL 25 auto-increment settings don't start at 1 (Case 35141)
- fixed — Report data broken when reporting user-time when using the embedded database (Case 35104)

© 2003-2011 SmartBear Software

- fixed — Migrating data 46 into Oracle failed when custom field title was blank
- fixed — Migrating file data from review dump put the content in the wrong place when the user overrides the location of the content cache
- fixed — Leaking database connections when "download files" command was used. Connections were reclaimed eventually but wasted resources.
- fixed — Perforce trigger truncated changelist description lines at '#' (Case 35271)

v2.1.719 — 06/18/07
- fixed — ccollab addchanges 406 fails for Subversion checkout at file system root (Case 35056)
- fixed — Windows command-line invocations not being parsed properly in certain special cases
- fixed — Database cleanup for exceptional case on multi-core servers where a bug in a previous build could cause a comment/defect to not be associated with a visible conversation, thereby making a review impossible to complete (Case 35074)
- fixed — Quoted strings in C-style languages could cause a line of code to be incorrectly colored in the side-by-side view if it contained escaped quotes (Case 35048)

v2.1.717 — 06/15/07
- added — Option to override the maximum file size for a review. (Case 32313)
- fixed — Perforce trigger 383 broken when files have been truncated due to excessive length. (Case 32313)
- fixed — Reverted files show up as "Reworked". (Case 34592)
- fixed — Files not contained in directories display incorrectly in review overview 152 screen.
- fixed — Remove spurious log messages about uploaded files not having previous version as this is the expected condition.
- fixed — Race condition prevents some pages from loading and causes others to only be partially loaded. (Case 35051)
- fixed — Admin license page 43 displays the wrong number of licenses.

v2.1.714 — 06/08/07
- fixed — Web page can take forever to load (or new comment takes forever to appear) when emails are being sent out due to the action just submitted (Case 34986)
- fixed — URL validation in input fields should check format but not actually attempt to connect (Case 34997)
- fixed — Perforce trigger 383 was blocking non-Collab-related changes to the changelist spec, specifically in the case of creating a new changelist from scratch (Cases 34436, 34812, 34915)
- fixed — ccollab addchanges 406 fails for Subversion checkout at file system root (Case 34992)
- fixed — Variable substitution 119 system could skip a variable if you strung together many variables in a row (Case 35002)

v2.1.713 — 06/06/07
- fixed — Drop-down items not displaying current values (Case 34963)

v2.1.712 — 06/05/07
- added — Drop-down typed custom fields 104 can now be given a default value
- added — ClearCase only: `ccollab addactivity` 229 now supports diffs against latest version from integration branch (Case 34382)
- fixed — Cannot add yourself 150 to a review if you're not already a participant or an administrator (Case 34834)
- fixed — Custom field description text not being displayed in the " create defect 170" form (Cases 34671, 34861)
- fixed — Binary file identification pattern 82 should not be case-sensitive (Case 34895)

v2.1.711 — 05/30/07
- added — Subversion server-side hook for automatically creating a review 410 whenever a change gets submitted

© 2003-2011 SmartBear Software

- added — Subversion server-side hook for uploading revision data 410 to the server after a change gets submitted
- added — Manual page describing when notifications 134 are sent to users
- added — More documentation and troubleshooting information about Perforce server-side triggers 383
- fixed — Database connection errors or `NullPointerException` errors in a race condition when sending email notifications
- fixed — Error running `ccollab addsvndiffs` 220 with Subversion 396 clients prior to v1.4.0.
- fixed — Scrollbar missing in chat 168 window for binary files (Case 34623)
- fixed — Do not display "0 changelists" under "local changes" when all changelists are in fact checked in (Case 34798)
- fixed — Broken links to manual pages on client tools when attaching materials 143 to a review
- fixed — Supplying a blank name for a role configuration causes problems in the administration GUI

v2.1.709 — 05/21/07

- added — Speed enhancements for the diff viewer 157 web view.
- added — ClearCase `ccollab addactivity` 229 support for diffs against integration branch (Case 34382)
- added — Subversion server-side hook 410 for ensuring that files are reviewed before they are checked in.
- added — User's Guide chapter for recommended hardware 62 and how to increase server speed 62.
- fixed — CVS variant of `ccollab addversions` 227 uploaded correct file content but incorrectly marked the versions as "added" in the web GUI (Case 34617)
- fixed — Subversion command-line can get too long under Windows with large numbers of files.
- fixed — SQL error trying to archive files using Oracle, SQL Server, or embedded database back ends.
- fixed — Subversion Eclipse plug-in for Eclipse v3.1 would fail to upload files if uncontrolled files were left in a project and the Subclipse preference "Select unversioned resources on commit" was enabled.

v2.1.708 — 05/16/07

- added — New Perforce trigger 383 for automatically updating changelist description with information about the associated review.
- added — Support for using wildcards with paths when using `ccollab addchanges` 216 with SCM system type `none` under the Windows `cmd` shell
- added — Support for Subversion's 396 `svn:externals` feature wherein a repository contains a soft link to another repository.
- added — Speed enhancements for the Review Summary 148 page when viewing very large reviews.
- added — Speed enhancements for common operations in the Subversion integration.
- added — "Obfuscated Dump 70" for both review-specific and whole-system database dumps to allow customers to send SmartBear tech support data dumps without sending any sensitive information.
- fixed — (Eclipse/Subversion plug-in) Changes from different resources but in the same repository were showing up in different changelists (Case 34544)
- fixed — The command-line client 196 could get confused when `svn switch` was used on a local check-out
- fixed — Eclipse Subversion plug-in can pick up the wrong repository configuration (Case 34399)
- fixed — Suppressed spurious (and harmless, but annoying) error messages we were printing to the Subclipse console
- fixed — URL's inside review comments can get mangled if they are very long (Case 34461)

v2.1.707 — 05/07/07

- added — Optimized subversion integration.
- fixed — Side by side not resizable when wrap lines turned off.
- fixed — Subversion integration did not properly handle added file in added directory.
- fixed — Recent participants list should only include participants from the selected workflow.
- fixed — System administrator account matched incorrectly when LDAP 50 configured.
- fixed — Content archive fails when using embedded database.

© 2003-2011 SmartBear Software

v2.1.706 — 05/01/07 **FIRST NON-BETA RELEASE OF v2.1**

- added — Documentation for SSL configuration 55.
- added — Author should get email notification when (a) Author is required to finish the review and (b) everyone else is finished, even if those other users are not themselves required to finish the review.
- added — Passwords entered on the command-line now echo asterisks (Java v1.6 only) (Case 34477)
- added — Eclipse Subversion integration can now add files to a review from the Synchronized view (Case 34442)
- added — Tutorial box about how reports 176 are cached and how to update them.
- fixed — `ccollab adddiffs` 219 failing to recognize a custom diff format (Case 34465)
- fixed — `ccollab addactivity` 229 adds wrong predecessor when multiple versions in activity aren't predecessors of each other (Case 32382)
- fixed — Subversion integration can upload HEAD version instead of HAVE in certain cases (Case 33456)
- fixed — Subversion integration threw exception when an uncontrolled file was added to a review (Case 34427)
- fixed — Subversion integration uploading incorrect file content when multiple parent directories, some not direct parents, are involved in the same changelist (Case 34328)
- fixed — Perforce integration throws exception if changelist state changes (Case 34308)
- fixed — Keyboard shortcuts 160 in the web user interface stopped working when the content view changed
- fixed — Switching roles on two participants in a review didn't reset the states of those users, which could result in reviews that cannot be closed and confusing Action Item messages
- fixed — Search box 174 should trim leading and trailing whitespace from text -- handles common case when pasting text from a web browser
- fixed — In brand new installs, if the license server cannot be reached users get a blank page after login.

v2.1.705 — 04/24/07

- added — Command line client needs to pick up Perforce changelist comments as title for new reviews.
- added — Command line client uses Subversion changelist comment as title for new reviews.
- added — Command line client uses Clear Case activity titles as review title for new reviews.
- fixed — Subversion exception adding an uncontrolled file to a review (Case 34427)
- fixed — Eclipse; Subversion arrows not appearing on select files page. (Case 34451)
- fixed — Reverted files do not show up properly if using different clients; Eclipse and command line.
- fixed — Invoking Windows commands with embedded spaces fails. (Case 34445)
- fixed — Keyboard shortcuts mask well known Windows shortcuts. (Case 34460)
- fixed — Custom fields can get into a state where changing a single custom field changes multiple fields (Case 33700)

v2.1.704 — 04/19/07

- added — Perforce trigger 383 to check that a review has been created, but not care if that review has been finished.
- added — Reinvite preference expanded to mean "works independently".
- fixed — Perforce GUI integrations 381 fail to find p4 after upgrading to 2.1.
- fixed — When migrating 46 from SQL Server 27 or Oracle 29 to MySQL 25, installer fails to properly configure user authentication. (Case 34353)
- fixed — Time-in-review database view updated to ignore time spent during "rework" phase.
- fixed — Specifying multiple files to addchanges fails for Subversion (Case 34354)
- fixed — Subversion added files showing up as modified (Case 34337)
- fixed — Search results should not return reviews that the current user does not have access to.
- fixed — Case sensitivity problem in Perforce trigger (Case 34288)
- fixed — Side by side version titles not appropriate for checked in changelists (Case 33422)

© 2003-2011 SmartBear Software

- fixed — Eclipse plugin subversion integration does not properly set the base revision (Case 34108)
- fixed — Eclipse plugin subversion integration does not handle spaces in filenames correctly (Case 34008)
- fixed — Subversion integration not finding reverted files (Case 34400)
- fixed — Uploading new changes to a review changes the state, but not the phase of the review.
- fixed — Action Item message for authors is misleading if the author is required to complete the review.
- fixed — External defects should be drawn with strike through and italic style (Case 34023)
- fixed — Externalized defect system should link to configured create defect URL (Case 34021)
- fixed — Remove usernames from Subversion URL's (Case 33425)
- fixed — Where possible, derive the author from the Subversion URL (Case 33806)
- fixed — Remove usernames from Subversion server config (Case 33807)
- fixed — Action item text did not always make sense for users whose activity is required to initiate phase change.
- fixed — Authors who marked reviews finished had no action item at all while waiting for other participants.
- fixed — Hide-show previous changelists must be sticky (Case 33888)

v2.1.703 — 04/11/07

- added — New role-specific option to have participants not re-invited to a review when another user makes a comment
- added — New role-specific option to control whether this role is allowed to change review details 149 while the review is active (Case 34302)
- added — Allowing edit of the review details 149 after the review is complete, but only when you're a participant or admin (Case 31478)
- added — Special warning when a user is viewing a review but is not a participant in that review (Case 34026)
- added — Display the title of the current workflow in the Review Overview 149 section (Case 33972)
- added — Now supporting either `ccollab set scm perforce` or `ccollab set scm p4` to configure 375 `ccollab` Perforce settings.
- fixed — Long comment text does not wrap in chat viewer. (Case 34289)
- fixed — CMVC integration fails when not all fields are populated
- fixed — Subversion was not properly handling file changes when parent directories were moved or deleted
- fixed — Subversion was not properly handling file changes when the parent path was the result of a directory-move, as with the first check-in on a new branch
- fixed — Some non-ASCII characters not displaying correctly when entered in the AJAX chat component 168 in side-by-side view (Case 34050)
- fixed — HTML characters in the review title cases Review Overview 148 page to display incorrectly (Case 34281)
- fixed — In ClearCase 322, `ccollab addactivity` 229 picks wrong latest version when not checked out (Case 33771)
- fixed — Email notifying a user that they have been added to a review didn't include their role (Case 34165)
- fixed — Reinstalling 182 command-line client sets SCM configuration to Perforce when Perforce is installed, even if the user specifically switched to a different SCM system (Case 33741)
- fixed — Confusing error message (`UnknownHostException`) printed to the console instead of a useful message (Case 33858)

v2.1.702 — 04/03/07

- fixed — NullPointerException getting defect custom field causes chat to stop updating (Case 33947)
- fixed — Externalized defects 171 should be drawn with a line through the text (Case 34023)
- fixed — Externalized defects should not hide other menu items (Case 34024)
- fixed — In-chat defect log on Review Summary 152 page does not show custom field 104 values (Case 34019)

© 2003-2011 SmartBear Software

- fixed — Deleting a defect from the Review Summary page does not confirm with dialog (Case 34018)
- fixed — Pending Perforce 366 changelist was not considered pending when `P4CLIENT` not specified
- fixed — Removed tutorial boxes in create review wizard when user preferences demand it
- fixed — CMVC 318 upload failed when a track had no parts

v2.1.700 — 04/02/07

- added — Eclipse support for Subversion (both pre-commit and post-commit)
- added — Eclipse support for ClearCase (pre-commit)
- added — SSL support 55 on web server and all clients
- added — Oracle 29 back-end database support
- added — Externalized defects 171 feature
- added — Keyboard shortcuts 160 in side-by-side view
- added — CMVC integration 318 (Case 33617)
- added — Special alert when a pop-up blocker prevents one of our web-application windows from opening (Case 33041)
- added — Added filenames to the "Defect List" report 176 (Case 33228)
- added — Full Java example file to the Java Client Library documentation
- added — Support for importing diffs 200 with inconsistent line endings (Case 33427)
- added — Support Perforce branch-add state (Case 26437)
- added — Command-line 196 now prints the review ID when a new one is created (Case 33344)
- added — Option to disable "system dump 46 " link for non-administrators
- added — Command-line 205 option `--nobrowser`
- added — New review wizard now displays the review title on all screens (Case 33750)
- added — Smarter algorithm for filenames with very long paths in side-by-side view (Case 33490)
- fixed — Don't include deleted files in the ZIP file 153 of review files.
- fixed — Underscores being clipped in word-wrapped view on certain browsers on certain platforms (Case 33699)
- fixed — Web GUI not allowing more than 128 characters for review title, but 255 characters are actually allowed (Case 33900)
- fixed — Deleted files fail to show deleted content (Case 33468)
- fixed — Perforce integration 366 was not honoring `$P4CONFIG` or the `p4 set -s` variables under Windows
- fixed — Notification when removed from review should come from user, not admin (Case 33354)
- fixed — Newly created defect custom fields show up as review custom fields (Case 33966)
- fixed — Proper default mono-space fonts for platforms lacking the "Courier New" font
- fixed — Perforce files opened for integrate should use head version as previous, not the have version (Case 33554)
- fixed — Comment and Defect icons not showing up properly when changelists were rolled up (Case 33621, 33676)
- fixed — Improved search / error-handling for external command-line tools
- fixed — Subversion upload was failing sometimes when a deleted file was in the changelist (Case 33689)
- fixed — SQL Server wasn't escaping characters correctly in certain `LIKE` queries
- fixed — `p4collab` 381 now contains a usage statement if you run it from the command-line
- fixed — Subversion integration 396 can now find `svn.bat` and other non-standard command-line clients
- fixed — Perforce commit prevention trigger 383 didn't honor the "canceled 149 " state for reviews.
- fixed — Custom field reporting views 432 can take so long to complete that the query times out (Case 33446)
- fixed — When review is deleted, activity records should be deleted as well (Case 33496)
- fixed — When uploading files from Subversion, ignore both `_svn` and the more common `.svn`.
- fixed — Cancel AJAX web requests when window closes; was causing browser to run out of connections (only 2 are ever allocated) and then hang (Case 33557)

© 2003-2011 SmartBear Software

- fixed — Web GUI URL auto-formatter is too greedy (Case 33636, 33907)
- fixed — Perforce `addchanges` trigger 383 can fail with no error message (33584)
- fixed — Error parsing ClearCase output when backslashes were used (33224)
- fixed — Viewing XML files sometimes shows SGML character entities instead of proper characters (Case 33280)
- fixed — Could throw NPE when Java's `mailcap` file is corrupt
- fixed — Could get database error when inserting too much text into a field
- fixed — No notifications when you cancel the review from the review planning phase (Case 33369)
- fixed — When you've exceeded your license 41 the system now allows the existing users to continue using the system and shuts out only the additional users, rather than shutting out *everyone* all together.

v2.0.621 — 04/09/07

- fixed — Allow editing of review data in complete phase. (Case 31478)

v2.0.620 — 04/02/07

- fixed — NullPointerException getting defect custom field causes chat to stop updating (Case 33947)

v2.0.619 — 03/26/07

- fixed — Web pages refer to external resources (`akamai.com`, `yimg.com`) (Case 33192, 33876, 33885)

v2.0.618 — 03/13/07

- fixed — `ccollab addchanges new .` 406 fails for Subversion 396 local changes (Case 33784)

v2.0.617 — 03/09/07

- fixed — Allow install as non root user on Unix systems. (Case 33622)
- fixed — Running `ccollab addp4diffs ...` 224 hangs on binary files. (Case 33581)
- fixed — Perforce trigger 383 never lets `ktext` files be checked in. (Case 33696)
- fixed — Running `ccollab addchangelist ...` 408 fails when there is a deleted file in the Subversion revision (Case 33689)
- fixed — Overall chat was missing unread flags. (Case 33633)

v2.0.612 — 03/01/07

- fixed — Perforce `addchanges` 383 trigger fails with no error message (Case 33584)

v2.0.611 — 02/28/07

- fixed — Running `ccollab set scmconfig p4 ...` 215 had no effect on Perforce integration.
- fixed — P4V integration 381 invoked Windows executable.
- fixed — Do not include deleted files in download (Case 33537)
- fixed — Null Pointer Exception when uploading file diffs (Case 33553)
- fixed — CVS diff parsing (`addcvsdiffs` 221) can fail if files have mixed line endings (Case 33427)

v2.0.609 — 02/21/07

- fixed — Uploading raw diffs 200 of binary files from Subversion could result in line-ending transformations
- fixed — Uploading raw diffs 200 with mixed line endings styles now works even if the diff generator program was not aware of the line-ending problem (Case 29801)
- fixed — Can get "Illegal Operation" exception trying to upload file content when only line-endings have changed (Case 33533, 33306)
- fixed — Subversion integration 396 fails to upload changes, either local or checked-in, when directory operations were involved (Case 33097)
- fixed — Subversion integration 396 fails to upload atomic changelists when one of the files is deleted
- fixed — In ClearCase 322, `ccollab addactivity` 229 was failing on directories (Case 33511, 33520)

© 2003-2011 SmartBear Software

- fixed — Perforce 366 branched adds not being added to the review (Case 26437, 32517)
- fixed — ClearCase 322 parsing of `\main\LATEST` can fail, and the selection of which version is "latest" was wrong in certain cases
- fixed — The `reviewcustom` and `defectcustom` database view 432 definitions were so inefficient that with a complex custom field 104 configuration the database can time out waiting for the query to complete (Case 33446)
- fixed — When a review is deleted, activity records 426 in the database are not deleted as well, leading to orphaned rows. This didn't affect the operation of the software but it could be confusing when making custom reports (Case 33496).

v2.0.608 — 02/09/07

- fixed — Perforce trigger 383 fails to ignore canceled reviews (Case 33402)
- fixed — On Windows, Subversion integration can fail to find `svn` client

v2.0.606 — 01/19/07

- added — Many more database views 432 for external reports 426
- added — Review overview 148 screen should not refresh automatically when the user is adding a comment or defect. The auto-refresh can cause the form fields to be erased (Case 25572)
- fixed — Server no longer requires external license server for user login 125
- fixed — Removing 150 unavailable reviewers should close the review (Case 32302)
- fixed — Overall chat 152 needs unread comment flags (Case 31491)
- fixed — Uncontrolled files prevent Review Display 157 from being shown
- fixed — Javascript error on Review Display 157 (Case 32202)
- fixed — User font preferences 127 ignored under certain browsers/platforms (Case 27991)
- fixed — When you submit Subversion change-set by ID with `ccollab addchangelist` 408, files that were deleted in the changelist are not uploaded at all. This does work when uploading files not yet checked in (Cases 29961, 30500, 33097)

v2.0.602 — 01/11/07

- added — Full support for `ccollab adddiffs` 200 and `addcvsdiffs/addsvndiffs/addp4diffs`
- added — Full support for `ccollab addversions` 227 for all SCM systems, not just ClearCase
- added — (Eclipse Plug-in) Pre-submit file uploading for Subversion now supported
- added — Option to disable changelist roll-up (Case 31485)
- fixed — Hide/show changelists not working when multiple SCM's in the same review
- fixed — Subversion executable not found causing command-line to fail (Case 32520)
- fixed — Subversion directory-add and directory-delete was causing the command-line client to fail to upload files
- fixed — Subversion protocol on `svn://localhost` URL's is now supported properly
- fixed — Author chat is automatically marked "read" if he uploads a new file version (Case 32702)
- fixed — Reports failing to execute under Microsoft SQL Server for lack of proper identifier escaping
- fixed — Installer incorrectly configuring context.xml for internal-based user authentication when running against Microsoft SQL Server
- fixed — Diff-caching algorithm was showing correct but inconsistent line-diff metrics depending on the order the files were uploaded
- fixed — Context-sensitive help wasn't able to jump down to a sub-section of a single page of the manual
- fixed — Horizontal scrollbar in side-by-side view under IE 7 with "no-wrap" enabled was sometimes not long enough to view the entire line (Case 30481)
- fixed — If server starts up before the database, the server will never recover (Case 13626)

v2.0.601 — 12/29/06

- added — new AJAX-based chat system really works like instant messaging (Case 24895)

© 2003-2011 SmartBear Software

- added — Microsoft SQL Server support (with migration path from MySQL)
- added — overall-review comment/defect chat area (Case 13622)
- added — server-side event-based trigger system for running custom scripts (Case 26886)
- added — link to download all review files to the local workstation (Case 24891)
- added — ability to cancel a review rather than deleting completely (Case 24892)
- added — ability to change participants while review is going on (Case 24894)
- added — database view for person-hours per review, user, and role
- added — command-line client options for overriding server URL, username, and password
- added — emails now include user's full name both in "to" and "from" fields (Case 23212)
- added — date/time of comments now displayed as a tool-tip (Case 10644)
- added — support for "local-mode" CVS servers (Case 25020)
- added — links to server/review debugging data from "System" screen
- added — user option for whether default side-by-side diff is "current vs. last upload" or "current vs. base version" (Case 25812)
- added — P4V/P4Win plug-in should list current reviews when attaching changelists to existing reviews (Case 28458)
- added — embedded database installer option for easy test servers (with migration path to MySQL or SQL Server)
- added — installer allows admin to set web server port number
- added — can create users with name and email address from the user administration page
- added — user administration page now shows which users logged in, inactive, disabled, and administrators
- added — deleted file content now viewable in side-by-side
- added — for SCM systems without atomic changelists, consolidate all changelists into a single list view
- added — for SCM systems without atomic changelists, rework uploads include reverted files
- fixed — insertion/deletion markers cause confusion; tooltips added to explain (Case 30393)
- fixed — filename different depending on which browser attached the external file (Case 25329)
- fixed — "accept" markers are now cleared on files that have been newly uploaded (Case 20534)
- fixed — with review in inspection phase, one reviewer is done but others aren't, no action item for the first reviewer (Case 25823)
- fixed — recent participant list included inactive users (Case 24141)
- fixed — user initials can be ambiguous (Case 24274)
- fixed — speed optimizations for very large reviews (Case 29003)

8.2.5 Version 1

v1.2.516 — 11/02/06

- fixed — Perforce trigger option for specifying a profile directory.
- fixed — Improve Perforce integration and trigger verbose logging.
- fixed — Installer removing attributes from realm declaration.

v1.2.515 — 10/26/06

- fixed — diff widget broken in Firefox 2.0 (Case 27312).
- fixed — not all files are rolled up into single changelist (Case 26821).
- fixed — preferences can fail to load client configuration (Case 27311).

v1.2.512 — 10/23/06

- fixed — attach uploaded changelists from the web ui (Case 26811).
- fixed — consolidated changelist view does not show all files (Case 26821).
- fixed — content archiving administration screen "hangs" with very large data caches (Case 26228).

v1.2.510 — 10/04/06

- fixed — Unable to mark conversations "read" when over 2000 comments present in the review (Case

© 2003-2011 SmartBear Software

25865).
- fixed — database connection leak.

v1.2.508 — 09/15/06
- added — options to only allow participants to view reviews/review content. (Case 24501)
- fixed — space used to mark insertion point was confusing. (Case 24246)
- fixed — whitespace inserted/removed at end of line causes rendered code to differ from real code. (Case 24767)
- fixed — syntax highlighting sometimes dropped spaces. (Case 25040)

v1.2.507 — 09/11/06
- added — automatically separates changed files from separate SCM systems
- added — administrative option to hide/show the option to display multiple changelists as a single "changelist"
- added — hide/show previous changes is sticky.
- fixed — clients were not honoring proxy settings taken from preferences
- fixed — increased maximum length of email message to 64k. (Case 24832)
- fixed — confusion of unique changelist ID's when using more than one Perforce server against a single PeerReview Complete database
- fixed — when displaying changelists in "single" mode, balls up files from different SCM systems instead of making one package for each SCM system

v1.2.506 — 08/24/06
- fixed — Changing mark reviews as fixed workflow setting had no effect.

v1.2.505 — 08/23/06
- fixed — Perforce changelists picking up wrong predecessor version (Case 24157).

v1.2.504 — 08/22/06
- added — Verbose logging for Perforce integration.
- added — Support for Subversion username/password supplied on command-line. (Case 24124)
- fixed — Downloading binary files in Internet Explorer broken. (Case 23671)
- fixed — Client installer failed to authenticate when changing server and user password. (Case 23535)
- fixed — Removed unnecessary AJP connector from server.xml file due to small known memory leak. (Case 23736)

v1.2.503 — 08/16/06
- fixed — More graceful handling of license server errors using cached license codes.
- fixed — Include documentation of how to configure non-proxied hosts.
- fixed — Increased number of characters allowed in certain meta-data fields.

v1.2.502 — 08/08/06
- added — URL's, email addresses, and issue ID's are now hyper-linked when displaying custom review/defect fields
- fixed — uploading local changes from CVS or ClearCase could pick up the wrong previous version
- fixed — Diff highlight expansion caused code to be not printed. (Case 23379)

v1.2.501 — 08/01/06
- fixed — Database error causes trigger to fail.

v1.2.500 — 07/28/06
- added — Administration screen for archiving old review contents.
- added — Option to add/remove/edit review and defect custom fields.
- added — Reviewers have the option of annotating the review materials before the review begins.
- added — Option to change notification email subject prefix
- added — Quick links on review creation wizard to add the current user to a review
- added — clients pick up default global configuration from `$HOME/.smartbear` and `$CWD/.smartbear`,

© 2003-2011 SmartBear Software

and do not create a `.smartbear` directory until preferences are actually saved.
- added — Perforce configuration value of "[none]" instructs the command-line utility to ignore the value completely
- added — ability to upload arbitrary diffs from CVS server
- added — line numbers in defect log
- added — selecting diff preferences causes immediate page refresh; don't have to click "Submit"
- added — file names now bold in changelist summary display
- added — option to show multiple changelists as a single unit
- added — user configurable email notification levels
- added — optional "create new user" form on login page
- added — three database views holding custom fields for reviews and defects, and extra user preference data
- fixed — First leading space is not displayed in side-by-side view (Case 10032)
- fixed — Normalize CVS paths from command-line and Windows client uploads (Case 11009)
- fixed — Status icons on review summary should link to the file (Case 10742)
- fixed — Accepted status cleared for newly uploaded versions (Case 10111)
- fixed — Chat notification icons do not update until side-by-side reloads
- fixed — Chat notification icons do not display when displaying single files
- fixed — Version selection and headings scroll off screen in side-by-side (Case 11213)
- fixed — Perforce GUI plug-in allows empty changelists to be uploaded to new reviews
- fixed — Only administrators should be allowed to change user logins (Case 8270)
- fixed — Complete line changes should be displayed as a delete followed by an add (Cases 10913, 10915)
- fixed — Review List report links linked to the wrong location (Case 11832)
- fixed — Word, Excel, and PDF documents show up as garbled text in side-by-side.
- fixed — Metrics by Review report failed to report defects per person-hour on small sample sizes.
- fixed — Metrics by Defect Type report failed to report opened per person-hour on small sample sizes.
- fixed — Next/Prev change buttons are disabled when there are no next/prev diffs.
- fixed — Clicking line of code jumps chat to correct area (Case 13280)
- fixed — Support filenames with adddiffs option (Case 13528, 14949, 16619)
- fixed — Diffs too greedy (Case 14152)
- fixed — No content uploaded for uncontrolled files (Case 14959)
- fixed — Perforce trigger has more explicit error messages when rejecting checkins. (Case 18398)
- fixed — Administrators always need access to review creation wizard. (Case 18399)
- fixed — Exported reports fail to open directly in Internet Explorer. (Case 13524)

v1.1.442 — 07/17/06
- fixed — Perforce trigger option for ignoring integration changelists (Case 14745).
- fixed — Perforce trigger added --verbose option for verbose logging (Case 16732).
- fixed — Perforce trigger sometimes compares wrong content (ensurecontentreviewed) (Case 16696).
- fixed — Perforce GUI integrations now include --verbose option for verbose logging.
- fixed — Perforce integration runs out of ports when large changelist present.
- added — Support bug system hyperlink in review Title
- fixed — Administrator account password update failed (Case 18606).

v1.1.436 — 05/22/06
- fixed — XML-RPC encoding broken on z/OS (Case 9367).
- fixed — NPE executing Clearcase cleartool (or any command line client executable) on Linux (Case 12781)

v1.1.435 — 05/17/06
- fixed — Perforce trigger should not run as submitting user (Case 13161).

v1.1.433 — 05/11/06

© 2003-2011 SmartBear Software

- fixed — Perforce usernames should not be considered case-sensitive.
- fixed — When using LDAP authentication, clients could not log in.
- fixed — Perforce trigger ensure content reviewed incorrectly rejects very large files.
- fixed — P4V/P4Win integration picks up environment variable for P4CLIENT.
- fixed — P4V/P4Win integration does not reject invalid changelists when creating new review.

v1.1.429 — 05/01/06
- fixed — Improve Perforce trigger error messages.

v1.1.428 — 04/26/06
- fixed — Perforce trigger ensure content reviewed misses additions/deletions from end of file (Case 11304).

v1.1.427 — 04/25/06
- fixed — Clear Case command line should only gather files from current view (Case 11799).

v1.1.426 — 04/21/06
- fixed — System administrator account lost administrator privileges under LDAP authentication (Case 11468).
- fixed — Command-line client not parsing Unix paths -- Clear Case integration (Case 11501).

v1.1.425 — 04/20/06
- fixed — Restored legacy ReviewList report.

v1.1.424 — 04/10/06
- fixed — Command line picked up wrong predecessor version when using ClearCase SCM (Cases 9040, 10738).
- fixed — Entering license code fails to update license.
- fixed — Client unable to locate cleartool on Windows.
- fixed — Reports not working on some headless Unix servers.

v1.1.423 — 04/05/06
- fixed — unattended installations preserve all configuration information.
- fixed — notifications include product name.
- changed — enabled unattended installations.

v1.1.422 — 04/04/06
- changed — disabled unattended installations until issues can be resolved.

v1.1.421 — 04/03/06
- fixed — author comments and trivial reviewer activity (e.g. accepting) were triggering erroneous emails to the author to come into the review
- fixed — when multiple already-submitted Perforce changelists with common files were added to a single review, the older ones were being hidden in "Previous Uploads" when they should be displayed along with the rest.
- fixed — user passwords no longer stored in the clear
- fixed — deleting defect severities or types causes UI glitches when old defects are displayed (Case 10268)
- fixed — side-by-side font wrong in some places (Case 10607)
- fixed — participants are not notified or re-invited to reviews when the author uploads a new set of changelists
- fixed — side-by-side windows should have the filename in their title
- added — support for stronger Perforce commit-trigger that checks whether the file list and file contents match between changelist and the review to make sure the developer didn't change anything between review-time and commit-time.
- added — user option for disabling syntax coloring in side-by-side view for faster content downloads
- added — option to create new review from command line client when running 'addchanges' or

© 2003-2011 SmartBear Software

'addchangelist'
- added — ability to diff previous file uploads in side-by-side view even when those uploads belong to different, discontinuous, already-committed changelists
- added — fixed defects are now indicated with a "green bug" to distinguish from still-open defects in the Review Overview screen
- added — in Review Overview screen, defects are now shown in the column of the user that first reported it rather than in a separate column
- added — added audit messages when a defect is marked fixed or open
- added — verbose option for command line clients to create debug logs
- added — more specific error message when attempting to upload an empty Perforce changelist
- added — display entire file path in defect log when the defect is linked to a particular file
- added — defect icon tooltips should include defect ID's for quick-reference
- added — command line clients prompt for password if not specified on command line so password is not in command history
- added — command line 'adddiffs' subcommand for reviewing differences between two locally accessible directories
- added — command line quiet option to suppress opening the text editor to edit file list
- added — cache control headers to allow caching of images, stylesheets, and JavaScript.

v1.0.410 — 03/20/06
- fixed — cannot close review with multiple reviewers when defects are entered directly into the defect log (Case 10116)
- fixed — proper error message inside P4V on changelists when uploading changelists that are not part of the current workspace
- fixed — Javascript error loading Defect Severities or Defect Types page

v1.0.409 — 03/16/06
- fixed — `ccollab syncusers` was failing when the Perforce user login differed from the Collaborator login only by case

v1.0.408 — 03/14/06
- fixed — notifications were no longer prefixed with PeerReview Complete

v1.0.407 — 03/14/06
- fixed — incompatibility with some versions of MySQL
- fixed — duplicate user logins possible when changing user login
- fixed — email notifications stop being sent after some time

v1.0.406 — 03/09/06
- fixed — memory leak with GUI client connections
- fixed — installer not setting VM heap size
- fixed — Case 9744 - NullPointerException in P4Win/P4V integration
- fixed — Case 9710 - All action items were showing urgent icon
- added — additioal logging in database code
- added — System Dump includes more VM information

v1.0.402— 03/02/06
- fixed — logout broken for some versions of Internet Explorer
- fixed — some clients unable to login
- fixed — logging now enabled by default at INFO level

v1.0.400 — 02/28/06
- added — chat conversations are now marked as "has chatted" and/or "as opened defect" and/or "has accepted" rather than the "undecided" and other confusing concepts
- added — new reporting subsystem with additional filters and export options
- added — added option to show "only uploaded version" even when other diffs are available

© 2003-2011 SmartBear Software

- added — better support for binary file uploads
- added — ability to make a user "inactive" (you can never delete users because they are needed for reports and to display old reviews)
- added — Action Items list now more specific about the exact state of the review as it relates to the viewing user
- added — e-mail notifications now more specific about the exact state of the review as it relates to the viewing user
- added — ability to jump from a defect in the review summary defect log directly into that point in the latest source code upload
- added — support for local temporary license code if the network is unavailable for on-line licensing
- added — new customer-accessible Java library to read/write everything in the Collaborator server remotely
- fixed — cannot "Complete Review" when there are unread comments on previously-uploaded changelists that have already been marked read in the currently-uploaded changelists
- fixed — when chat is carried forward from older uploads, chat icons on review summary aren't being displayed
- fixed — cvs rlog command reports error with certain versions of the CVS client
- fixed — cannot delete a review when database is in a certain rare state
- fixed — "out of memory" errors for certain operations
- fixed — database connection leak with certain types of authentication
- fixed — correct diff but erroneous intra-line highlighting when unchanged text is symmetrical on either side of change
- fixed — selection highlighter highlights wrong line of code when window is narrow

v1.0.361 — 02/09/06
- added — (Case 8439) support for more types of CVS repository specifications.

v1.0.360 — 02/08/06
- added — (Case 8941) support for trial license codes which do not contact license server.

v1.0.359 — 01/27/06
- added — (Case 8486) when licensing server cannot be reached, retry on backup port.

v1.0.358 — 01/24/06
- added — (Case 8486) support for HTTP proxy authentication when validating license codes.

v1.0.357 — 01/20/06
- added — (Case 8420) command line client support for uploading Subversion revisions.
- fixed — (Case 8439) in certain cases, command line client was failing to upload CVS changes.

v1.0.356 — 01/09/06
- fixed — (Case 8319) some client upgrades failing with authentication issues
- fixed — (Case 8344) improved error message when server failed to contact license server.

v1.0.355 — 01/03/06
- fixed — clients were not properly authenticating when using LDAP

v1.0.354 — 12/29/05
- added — option to allow system administrator to participate in reviews
- fixed — (Case 7560) unable to upload files after clean install with old database.
- fixed — LDAP role fields were confusing and not required for basic LDAP support.

v1.0.353 — 12/21/05
- added — progress indicator for when loading file contents in file viewer
- fixed — (Case 7530) server failed to get user initials in some cases

v1.0.352 — 12/19/05

© 2003-2011 SmartBear Software

- fixed — (Case 7501) Stale cookies were causing issues with login/logout.

v1.0.351 — 12/12/05

- added — (Case 6640) previously-uploaded changelists are now hidden by default to avoid confusion when verifying fixes
- added — (Case 6812) link to leap from defect into associated file/line
- added — (Case 6871) support more than one file open at a time, bringing forward side-by-side window if already open
- added — (Case 6824) automatically hyperlink URL's and e-mails in changelist comments, chat comments, defect text, and review overviews
- added — quick-report of recent changelists and unreviewed changelists when attaching materials to a review
- added — application level log system
- added — support for LDAP authentication
- added — new installer prompts for key system parameters; server configuration split between server.xml and context.xml
- fixed — (Case 7226) code displayed in word-wrapped mode now doesn't have extraneous whitespace inserting into long tokens
- fixed — (Case 6932) not able to upload any changelist if Perforce default changelist was not empty
- fixed — (Case 6949) downloaded reports were not sending a filename
- fixed — added special error message when uploading a pending Perforce changelist that belongs to the current P4USER but to a different P4CLIENT
- fixed — added `java.library.path` to the local application scan directory for systems without environment variables
- fixed — uploading additional Perforce changelist didn't update the time-of-upload; was taking the time-in-changelist instead
- fixed — uploading older Perforce changelist was sometimes destroying metrics data for previous file versions
- fixed — not able to upload submitted Perforce changelists from the command-line when P4CLIENT environment variable is changed
- fixed — after following a link but not being authenticated, after authentication you go to the home page instead of the linked page

v1.0.349 — 10/25/05

- added — (Case 6595,6804) side-by-side difference viewer needs "ignore whitespace" and "ignore capitalization" options
- added — support for client-side HTTP proxies
- added — link to download server logs in "System" debugging page
- fixed — (Case 6780) Perforce server-side trigger should ignore changelists consisting of branched paths only for purposes of both review and automatic uploading
- fixed — (Case 6814) error installing client under Windows when user-level PATH variable doesn't already exist

v1.0.347 — 10/14/05

- added — "mark all comments read" quick-button in side-by-side view; administrators can disable this feature
- fixed — unable to "mark read" on new chat if already marked read in the same chat session
- fixed — issue regular expression was case-sensitive

v1.0.346 — 10/10/05

- added — new Action Item when e-mail or full name not currently set for a user
- added — prompt to download local client software from the server
- added — administrative contact information is now required

v1.0.345 — 10/03/05

© 2003-2011 SmartBear Software

- added — (Case 6664) ability to view files/metrics for uploaded changelists in "Attach Materials" section of the New Review Wizard
- added — (Case 6652) Action Items with no "new" comments should be marked as such on the "Action Items" list, and should be low priority
- added — (Case 6666) refresh button for chat window
- fixed — (Case 6635) URL's in e-mails give error when opened under Eudora
- fixed — (Case 6661,6643) JavaScript error with next/previous buttons in side-by-side view
- fixed — entering non-digits in the "jump to review" menubar field results in strange error message
- fixed — file paths with backslashes not word-wrapping on Review Overview screen

8.2.6 Version 0/Alpha

v0.9.344 — 09/26/05

- added — initial support for ClearCase
- added — initial support for the Windows GUI Client
- fixed — (Case 6621) "add defect" comment field still limited to 255 characters in form on Review Overview page
- fixed — (Case 6475) "edit defect" command should update GUI widgets to indicate "edit" rather than "create"

v0.9.343 — 09/21/05

- added — command-line support for uploading CVS changes by label
- fixed — (Case 6593) error uploading Perforce files: "unrecognized chunk"
- fixed — (Case 6540) bogus date in user admin screen for users who have never logged in
- fixed — (Case 6475) "edit defect" should have different form text than "create defect."

v0.9.342 — 09/15/05

- fixed — (Case 6544) script error viewing diffs in certain files, or when file content is not yet fully loaded
- fixed — action items for "waiting for comments" should appear after "respond to comments" because they are lower priority
- fixed — syntax highlighting split by intra-line difference causes incorrect colors in both difference and syntax display

v0.9.341 — 09/14/05

- added — (Case 6471) external base server URL (used with e-mail notifications) is now configurable by the administrator
- added — (Case 6408) participant-picker needs recently-used list to speed up the assignment process
- fixed — (Case 6508) 255 char limit in overview
- fixed — (Case 6529) error attempting to send email with a certain JVM
- fixed — (Case 6534,6538) typing comment then clicking "Accept" causes comment to be lost
- fixed — (Case 6522) `ccollab addchanges` cannot handle relative paths

v0.9.339 — 09/09/05

- added — (Case 6267) **workflow revamp: faster to "accept," concepts of "reject" and "defect" have been fused, no more accept/unsure/reject determination for commentary, comments carry forward to code verification step, defects shown in side-by-side next to comments for editing and marking fixed**
- added — (Case 6372) large text fields should expand as more text is entered into them
- added — (Case 6406) administrative option to set system-wide tab width for source file display
- fixed — (Case 6406) tabs not being displayed properly in side-by-side view
- fixed — (Case 6355) should not be allowed to add a changelist to a review that is completed or that you're not a participant in or a creator of
- fixed — (Case 6265) comment text should not be limited to 255 characters

© 2003-2011 SmartBear Software

- fixed — (Case 6382) client error message not helpful when incorrect server/port is specified, especially when a real service is present
- fixed — (Case 6354) clicking on a specific command-line doesn't open the file to that line

v0.6.337 — 08/24/05
- fixed — (Case 6275,6279,6281) database errors accessing user data
- fixed — (Case 5624) error not able to locate `content-cache` subdirectories on the server

v0.6.336 — 08/23/05
- added — initial support for CVS integration
- added — (Case 5628) user-preference to set number of lines of context displayed in side-by-side view
- fixed — error accessing reviews when review meta-data is missing from the database
- fixed — (Case 6252) exception adding comments to the comment-list

v0.6.335 — 08/19/05
- added — (Case 5551) launch browser window after P4Win plug-in attaches changelist
- fixed — (Case 5528) error uploading large file
- fixed — (Case 5615) incorrect LOC-changed metrics for file uploads
- fixed — (Case 5617) extra whitespace in side-by-side view of code when tabs are used in a certain way
- fixed — (Case 5618) file download should supply filename to the browser for better open/save handling on the browser end

v0.6.334 — 08/18/05
- added — (Case 5553) ability to alter review overview information during any phase of the review (without reverting back to "Planning" mode)
- added — (Case 5612) reviews that just entered the Inspection phase (either because just created or because fixes are being verified) should be displayed as "Waiting for comments" on the author side until a comment is actually made; the author should be notified by e-mail when this comment is made.
- fixed — not properly HTML-escaping certain characters in review titles
- fixed — (Case 5600) not always receiving e-mails as review in a new code review or when fixes have been uploaded
- fixed — (Case 5613) exception when accessing admin screen when upgrading from a certain database version

v0.6.332 — 08/17/05
- added — (Case 5590) ability to download file versions directly from the side-by-side view
- added — (Case 5576) optional feature to send e-mails from the review creator's e-mail address rather than the system default from address
- added — (Case 5387) gutter icons in side-by-side show which lines have associated chat comments
- fixed — (Case 5528) error uploading large file
- fixed — (Case 5577) uploaded "previous version" is actually the HEAD revision instead of the last-synched revision
- fixed — (Case 5584) erroneous error message while uploading files from a Perforce branch
- fixed — (Case 5386) skip-lines algorithm in side-by-side view should treat chat comments as significant lines, not just changes.
- fixed — (Case 5597,5598) incorrectly-encoded SGML character entities when intra-line diff splits on an encoded character

v0.6.331 — 08/14/05
- added — user preference for flipping the side-by-side view to older-on-right instead of the default older-on-left
- added — (Case 4388, 5279, 5424) ability to compare follow-up change against other uploaded changes in addition to the original SCM base version

© 2003-2011 SmartBear Software

- added — (Case 5335) participant input list in Review Creation Wizard automatically adds additional rows for large number of participants
- added — (Case 5544) better e-mail headers including easily-filterable text and the name and ID of the related review
- added — (Case 5468) drill-down from Review List report to actual review view
- added — (Case 5426) new report: Review List with Perforce Changelists
- added — (Case 5567) option to disable "issued fixed" throughout the application
- fixed — (Case 5550) lines with trailing whitespace are showing up with false-positive differences
- fixed — (Case 5484) errors on changelist-upload when no bug system regular expression is given by the administrator
- fixed — (Case 5515) new defects opened on completed reviews should change the review phase to Phase II: Review
- fixed — (Case 5557) erroneous exception when uploading file data under a special condition

v0.6.330 — 08/10/05

- added — global configuration options to ease per-user installation
- fixed — PATH variable not being set as environment-expand registry value under Windows
- fixed — (Case 5478,5482) uploading certain Perforce changelists takes exceedingly long time
- fixed — (Case 5481, 5516, ...) erroneous ccollab error requiring review-id and changelist-id when files are in the default changelist
- fixed — (Case 5524) better error messages when configuration files are not accessible
- fixed — (Case 5508, 5525) better error message when `http://` is missing from collaborator server specification

v0.6.327 — 08/08/05

- added — (Case 5426) ability to jump to review by ID from the menubar
- added — (Case 5471) command-line should pick up Perforce configuration from environment variables
- fixed — (Case 5453, 5454) exception working with changelist from uploaded file
- fixed — (Case 5460) exception viewing certain review detail reports
- fixed — (Case 5476) javascript error auto-refreshing review page under certain IE version
- fixed — (Case 5479) word-wrapping is breaking inside SGML character entities on side-by-side source code view

v0.6.325 — 08/07/05

- Case 5428 — added — support for PHP-style review logging data for older upgrade paths
- Case xxxx — added — report data can now be retrieved from the command-line in any format, with optional filters
- Case xxxx — fixed — PHP-upgrade path fixes

v0.6.323 — 08/05/05

- Case 5366 — added — now collecting time-spent-in-review metrics per user, per role, per review, per phase, per defect count/type/severity
- Case xxxx — added — report data can now be exported in XML or CSV format
- Case xxxx — added — report data can now be retrieved from the command-line in any format, with optional filters
- Case 5393 — fixed — administrator should not be allowed to change login name
- Case 5393 — fixed — user preference title has wrong username in title when administrator edits other users' information

v0.6.322 — 08/01/05

- Case 5381 — added — configurable review participant role behavior to support different workflows
- Case xxxx — added — ability to delete reviews when they're still in planning phase

v0.6.321 — 07/29/05

- Case 5278 — added — P4Win/P4V integration for creating a new review using a changelist

© 2003-2011 SmartBear Software

- Case xxxx — added — client installer now picks up Perforce configuration automatically (can still be overridden from the command-line)
- Case xxxx — added — client installer now prompts for and verifies server connectivity settings (can still be overridden from the command-line)
- Case xxxx — added — new "Licensing" administration page
- Case xxxx — added — administrative settings: minimum allowable build numbers for command-line and windows clients to force users to upgrade

v0.6.320 — 07/26/05

- Case 5278 — added — P4Win/P4V integration for uploading changelist data and associating it with an existing review
- Case 5342 — fixed — cannot switch participant roles in Review Planning
- Case 5344 — fixed — new review created; next/prev buttons work but links on the wizard pages list on left don't work
- Case 5345 — added — server system parameters on the System debugging link
- Case xxxx — fixed — spurious error when uploading already-committed changelists when the changelist has already been uploaded

v0.6.319 — 07/25/05

- Case xxxx — fixed — error sending e-mails with older Java mailer
- Case xxxx — fixed — associating changelist with review not properly encoding certain string before inserting into database
- Case xxxx — fixed — error-handling, validation, and help text for bug-tracking integration items
- Case xxxx — added — synchronizing Perforce userlist with Collaborator from the command-line

v0.6.317 — 07/25/05

- Case 5041 — added — uses value of REMOTE_USER for automatic log-in when password is blank
- Case xxxx — fixed — error accessing new user preference item

v0.6.316 — 07/21/05

- Case 4614 — fixed — new wizard prompts user better about pre-checkin changes
- Case xxxx — fixed — error where "mark as read" while also changing comment status doesn't get reflected on review summary page
- Case 5280 — added — user preferences for side-by-side word-wrapping, and font family and size
- Case xxxx — added — many more Web Service API's

v0.6.314 — 07/15/05

MAJOR UPGRADE! This new alpha release sees the unveiling of the new PeerReview Complete platform. We've switched from our proof-of-concept PHP platform to an industrial-strength Java platform with greatly enhanced scalability, support for multiple databases, and an improved, simplified workflow.

v0.4.215 — 06/14/05

- Case 4614 — fixed — new wizard prompts user better about pre-checkin changes
- Case 4607 — added — test PHP configuration at install-time
- Case ???? — added — optimizations for database communication cutting page-load time in half

v0.4.213 — 06/01/05

- Case 4562 — fixed — resizing window causes JavaScript error (IE only)
- Case 4564 — fixed — maximizing window causes JavaScript error (IE only)
- Case 4567 — fixed — scroll position is no longer remembered on screen refresh (IE only)
- Case 4563 — fixed — word-wrap still wraps sometimes (IE only)

v0.4.212 — 05/30/05

- Case ???? — fixed — all HTML and CSS now validated HTML 4.0.1-Transitional
- Case 4144 — fixed — focus change in accept/reject/undecided combo-box doesn't move selection

© 2003-2011 SmartBear Software

rectangle
- Case 4126 — fixed — user's file-diff preferences not persisted (i.e. lines of context, ignoring case)
- Case ???? — fixed — line-selection rectangle obscures underscore characters in certain browsers
- Case ???? — fixed — word-wrapping algorithm in file-diff too pessimitic, especially under IE
- Case ???? — fixed — tightened up intra-line difference highlighting
- Case ???? — added — optional new line-comparison frame makes it easier to compare long lines
- Case 4126 — added — option to disable word-wrapping in file-diff file content
- Case 4126 — added — buttons to advance to the next/previous change
- Case 3300 — added — review-create phase is now a "wizard"
- Case 3644 — added — on-line licensing system
- Case 4316 — added — new report on number of comments made per review / per user
- Case ???? — added — ability to review whole files and file-differences uploaded independantly from version control

v0.4.211 — 04/21/05
- Case 3858 — fixed — sometimes vertical scrollbar not appearing in file-diff view
- Case 3858 — added — side-by-side divider between chat and file content is now move-able so you can easily "hide" comments while reading a file
- Case 3949 — added — ability to jump to a particular review by ID from the menubar

v0.4.210 — 04/20/05
- Case 3929 — fixed — disallow running the command-line client against the Code Collab demo server (the default configuration)
- Case 3930 — fixed — javascript error when planning a review and the review is not yet ready to proceed to the next stage
- Case 3920 — added — ability to delete changelist associations from reviews
- Case ???? — added — "revert" button on various forms

v0.4.209 — 04/19/05
- Case 3905 — fixed — pending changelists shouldn't be available for adding from the website
- Case 3677 — added — "Save" buttons on forms remember scroll position for easier data-entry on long pages
- Case 3721 — added — administrative usage monitor available from bottom of admin "User" page
- Case 3838 — added — ability to set minimum build number for the command-line client from the Collab server admin settings page
- Case 3842 — added — issues mentioned in changelists should be associated with the review automatically
- Case ???? — added — reports can now "group by" certain things (see "Assignments" and "Reviews" for examples)

v0.4.208 — 04/12/05
- Case 3687 — fixed — **files in "add" state in Perforce changelist not being picked up by the command-line client**
- Case 3563 — added — user administration page: add users manually; view info and usage; admins can set "admin" flag and other user preference informatiion for other users
- Case 3702 — added — customization administration page: change text for defect severities, defect types, and phase titles and prompts
- Case 3694 — added — "Remember Me" should redirect the login page to the user's review home page.
- Case 3697 — added — report enhancements: review-created date on report and summary page; complete user comment table on review summary page
- Case 2623 — added — "Preview Changes" on changelist during Review Planning phase
- Case 3664 — added — margin icon for comments is now a pencil instead of a meaningless arrow
- Case 3681 — added — help documentation on the purpose and behavior of user roles
- Case 3699 — added — phase prompt bullets are hidden when tutorial mode is disabled

© 2003-2011 SmartBear Software

v0.4.207 — 04/11/05

- Case 3617 — fixed — local changes not showing up in side-by-side view
- Case 3606 — added — review phase data to getReports(), so scripts can check for review-completion, not just review-existance
- Case 3614 — added — "Action Items" list on the home page; daily e-mail action items reminder (if any are present)
- Case 3616 — fixed — observers should not prevent a review from proceeding to the next phase
- Case 3645 — added — removed "controller" role
- Case 3618 — added — ability to move from Phase II back to Phase I
- Case 3622 — fixed — prevent moving from "Planning" to "Reviewing" when there are unsaved changes to the Participant List
- Case 3623 — added — support "Remember Me" log-in option to make logging in easier
- Case 3660 — added — adding changelist to the review should automatically add the changelist author to the review in the "Author" role unless that user is already added in a different capacity
- Case 3662 — fixed — commenting on new line of code with "Accept" without additional commentary didn't work
- Case 3665 — fixed — inconsistent wording: "neutral" vs "undecided"
- Case 3666 — fixed — review in Phase II should show comments from all participants, not "consensus." Cannot really show "undecided" or "unreviewed," only what comments are made
- Case 3676 — added — usernames always get initials; smarter algorithm for pulling out initials automatically

v0.4.206 — 04/08/05

- Case 3471 — added — all users imported from Perforce, not just those in recent version history

v0.4.205 — 04/08/05

- Case 3606 — added — ability to call arbitrary RPC methods from the command-line
- Case ???? — added — help documentation on XML/HTTP/RPC integration points

v0.4.203 — 04/05/05

- Case 3557 — fixed — author needs notification when review enters "Rework" or "Completed" phase
- Case 3558 — fixed — shouldn't be able to close a review when "new" comments are pending
- Case 3559 — fixed — creating new users from the front page results in accounts that cannot be logged in to
- Case 3560 — fixed — install should check for writable `/file-cache` directory; instructions should say to check this also

v0.4.202 — 03/29/05

- Case 3474 — fixed — error message when incorrect file data is shown
- Case 3485 — fixed — add-defect form had incorrect maximum limit on number of lines in the file
- Case 3486 — fixed — Perforce not loading previous file versions (reload version control data using "Clear Version Control Data" link in `install.php`)
- Case 3470 — added — can now delete existing review from the bottom of the "Review Summary" tab
- Case 3472 — added — passwords in database are now encrypted

v0.4.201 — 03/28/05

- Case ???? — fixed — not compatible with MySQL v3.23

v0.4.198 — 03/24/05

- Case 3316 — fixed — `ccollab synchscm` documentation is confusing
- Case 3396 — fixed — deletions not handled properly in Perforce
- Case 3317 — added — option to disable "create new user" from front page
- Case 3367 — added — debugging support in server and command-line utility
- Case 3369 — added — added support for content-compression for browsers that support it

© 2003-2011 SmartBear Software

v0.4.186
First alpha release

8.3 Appendix C: Java VM Options

JVM options can be configured in `.vmoptions` files installed in the same directory as any of the PeerReview Complete Java applications. The name of the file must correspond to the name of the executable file, e.g., `ccollab-server.vmoptions` for the server, `ccollab.vmoptions` for the command line client, or `ccollabgui.vmoptions` for the GUI client. This chapter summarizes some of the more common options used.

Note: The last line in a `.vmoptions` file must be followed by a new line.

To specify Java system properties that will be available to the application, use this syntax:

```
-Dvariablename=value
```

PeerReview Complete Server Properties
The following Java system properties are specific to the PeerReview Complete server.

Property Name	Purpose and Usage
`smartbear.userauth.class`	This property specifies the authentication adapter used to authenticate users. Under normal circumstances, this property need not be set. Do not set this property unless directed by SmartBear technical support.
`smartbear.rpc.session.timeout`	The timeout, in seconds, for XML-RPC sessions. The value must be a positive integer. Default is 30.
`com.smartbear.ccollab.binary.converter.threads`	The maximum number of threads allocated to convert binary documents into images for review. This value must be a positive integer. Default is 4.
`com.smartbear.ccollab.license.noperiodicupdate`	If set to a non-blank value, this property prevents the PeerReview Complete server from checking for license updates.
`smartbear.appstate.period`	The delay, in milliseconds, between runs of the application state recorder. The state recorder periodically writes information about the state of the application into the database for future trend analysis. Default is 15 minutes.
`smartbear.appstate.startup.delay`	The delay, in milliseconds, before the application state recorder starts. Delaying the application state recorder allows the application initialization process to complete before anything is written to the database. Default is 5 minutes.
`smartbear.chat.request.limit`	The maximum number of chat update requests allowed per minute. This value will be used for active reviews and the clients and servers will automatically scale back the request rate as activity on the review declines.

© 2003-2011 SmartBear Software

`com.smartbear.database.longquerythreshold`	The time in milliseconds to allow database queries to run before logging a warning message. This is used to help debug bottlenecks and characterize the behavior of complex queries such as those used by the custom reports system. Default is 333 milliseconds.
`com.smartbear.diff.cache.maxentries.memory`	The number of entries allowed in the diff cache. The size of any given entry can vary significantly, but this tuning parameter provides a rough mechanism for tuning the cache size. Default is 2000.
`com.smartbear.lines.cache.size`	The maximum size of the parse lines cache. Values should be a memory size in kilobytes or megabytes indicated by the case-insensitive suffixes "k" and "m" respectively. For example, 20 megabytes could be indicated by "20m" or "20M". Default value is 5 megabytes.
`com.smartbear.syntax.cache.size`	The maximum size of the syntax highlight cache. Values should be a memory size in kilobytes or megabytes indicated by the case-insensitive suffixes "k" and "m" respectively. For example, 20 megabytes could be indicated by "20m" or "20M". Default value is 50 megabytes.
`com.smartbear.ccollab.notification.max.retry.interval`	The time (in milliseconds) to wait between attempts to connect to the SMTP server.

Command Line Client Properties
The following properties are specific to the Command Line Client.

Property Name	Purpose and Usage
`smartbear.ccollab.upload.truncate.size`	The size, in bytes, to truncate uploaded files. This threshold prevents clients from inadvertently overloading the server with files too large to process. The default value is 4MB.

Java Heap Size
The default maximum heap size available for PeerReview Complete client applications is 64MB. PeerReview Complete Server maximum heap size is set to 1GB by default. The maximum heap size can be increased with the `-Xmx` switch. For example, to set the maximum heap to 256MB, use the following line:

```
-Xmx256M
```

`G`, `M` or `K` can be used to represent gigabytes, megabytes or kilobytes. Note that not all JVM's support `-X` options.

Network Proxy Settings
Server proxies can be configured in the `ccollab-server.vmoptions` file. See Network Configuration 39 for the settings.

© 2003-2011 SmartBear Software

Network Connection Debugging

For debugging LDAP or SSL connection problems, or other network related issues, an additional server log file with trace information, `output.log`, will be created at restart with the following line in `ccollab-server.vmoptions`:

```
-Djavax.net.debug=all
```

© 2003-2011 SmartBear Software

Index

- A -

account management 124
AccuRev 288
action 79
action items 139
ActiveDirectory 50
administration
  archiving data 122
  bug tracking integration 73
  custom fields 104
  display options 73
  email configuration 99
  general settings 73
  roles 107
  system status 123
  triggers 115
  user management 84
  variable substitution 119
  workflow 112
archiving files 122
Authentication
  ActiveDirectory 50
  LDAP 50

- B -

backup/restore 46
bug-tracking integration 73, 442

- C -

chat 157
chat controls 167
ClearCase 322
ClearQuest 322
CMVC 318
commit 79
configuration 73
contact information 24
content 44
content cache 44
cookies 125
creating reviews 140
custom fields 104
CVS 302

- D -

database
  backup 46
  connectivity 31
  embedded 25
  Hypersonic 25
  installing 24
  Microsoft SQL Server 27
  migration 46
  MySQL 25
  Oracle 29
  restore 46
  SQL Server 27
  upgrading 37
  zero-configuration 25
defects
  creating 157, 170
  custom fields 104
  defect log 148
  deleting 157
  editing 157
  externalizing 171
  marking fixed 148, 157
  verification 171
  workflow 112
diffs 198, 200
display options 73

- E -

email configuration 99
Enterprise Organization 140
external issue-tracking integration 442

© 2003-2011 SmartBear Software

- F -

features 1
file content 44
file view 157
finding reviews 174, 176
fixed seat licensing 41
floating seat licensing 41

- G -

grace seats 41
Groups 88
GUI Client 190

- H -

help 69
HTTPS 55

- I -

installation
server component 31
integration
server-side triggers 115
integrations 426
AccuRev 288
ClearCase 322
ClearQuest 322
CMVC 318
CVS 302
external issue-tracker 442
Perforce Plug-in 336, 366
SourceSafe 356
Subversion 396
Team Foundation Server 349
TFS 349
UCM 322
Vault (SourceGear) 390
issue-tracker integration 442

- J -

JMX monitoring 60

- K -

keyboard shortcuts 160
known issues 460

- L -

LDAP 50
licensing 41
logging in 125
logging out 125

- M -

metrics 176
analysis 456
defect density 456
defect rate 457
definitions 455
inspection rate 457
reporting 427
Microsoft
SQL Server 27
migration 46
monitoring 60
Moving Parts 1
MySQL 25

- N -

network configuration 39

- O -

Oracle 29

© 2003-2011 SmartBear Software

- P -

P4V 381
P4Win 381
Perforce
  overview 336, 366
  P4V 381
  P4Win 381
  server triggers 383
  triggers 383
phone number 24
platform notes 40
pre-commit 79
Print to Review 282
proxies 39

- R -

redact 173
releases, software 462
reporting 174, 176, 427
reverse proxies 39
review
  overview 136
reviews 136
  action items 139
  by file 174
  by participant 174
  canceling 148
  creating 140
  custom fields 104
  file view 157
  metrics 176, 455, 456
  participants 140
  recent 174
  reporting 176, 427
  searching 174
  side-by-side view 157
  summary screen 148
  uploading files 140
  workflow 112, 140
roles
  configuration 107
  in reviews 140

- S -

screens
  action items 139
  admin, archiving files 122
  admin, bulk email 99
  admin, custom fields 104
  admin, email configuration 99
  admin, general settings 73
  admin, licensing 41, 84
  admin, roles 107
  admin, system status 123
  admin, triggers 115, 119
  admin, user management 84
  admin, workflow 112
  create review 140
  file view 157
  home 125, 139
  installer, command-line 182
  installer, server 31
  login 125
  preferences 127
  reporting 176
  review create 140
  review summary 148
  searching 174
  side-by-side view 157
  system 46, 69
  user preferences 127
scripting 115
searching 174, 176
security 55
Server Component
  databases 24
  installation 31
  overview 24
  technical specifications 61
  upgrading 37
server-side triggers 115
service 40

© 2003-2011 SmartBear Software

settings 73
shortcuts, keyboard 160
side-by-side view 157
software releases 462
SourceGear Vault 390
SourceSafe 356
SQL Server 27
status report 123
substitution of variables 119
Subversion 396
summary screen 148
syntax coloring 161
system status 123

- T -

Team Foundation Server 349
TFS 349
Tray Notifier 278
trials 25
triggers, server-side 115
troubleshooting
  command-line client 204
  database 46
  known issues 460
  network 39
  platform-specific 40
  server component 69

- U -

UCM 322
uploading
  content 198, 200
  diffs 198, 200
User Account Control 283
user account management 124
user management 84
user preferences 127
users
  action items 139
  administrators 84
  creating 84, 125
  grace 41
  licensing 41
  list 84
  logging in 125
  logging out 125
  preferences 127

- V -

variable substitution 119
Vault 390
version history 460, 462
Visual SourceSafe 356

- W -

Windows
  service 40
workflows
  configuration 112
  custom fields 104
  in reviews 140
  roles 107

© 2003-2011 SmartBear Software